



UNIVERSITAS INDONESIA

**PEMANTAUAN KEPATUHAN SECARA OTOMATIS MELALUI
ANALISIS LOG PADA MOODLE BERBASIS KECERDASAN
BUATAN**

SKRIPSI

**YAN CHRISTOFER SILALAHI
2106752464**

**FAKULTAS FAKULTAS ANDA
PROGRAM STUDI ILMU KOMPUTER**

**DEPOK
BULAN TAHUN**



UNIVERSITAS INDONESIA

**PEMANTAUAN KEPATUHAN SECARA OTOMATIS MELALUI
ANALISIS LOG PADA MOODLE BERBASIS KECERDASAN
BUATAN**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Gelar Jurusan Anda

**YAN CHRISTOFER SILALAHI
2106752464**

**FAKULTAS FAKULTAS ANDA
PROGRAM STUDI ILMU KOMPUTER**

**DEPOK
BULAN TAHUN**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Yan Christofer Silalahi

NPM : 2106752464

Tanda Tangan :

Tanggal : Tanggal Bulan Tahun

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Yan Christofer Silalahi
NPM : 2106752464
Program Studi : Ilmu Komputer
Judul Skripsi : Pemantauan Kepatuhan secara Otomatis melalui Analisis Log pada Moodle Berbasis Kecerdasan Buatan

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Ilmu Komputer, Fakultas Fakultas Anda, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing 1 : Amril Syalim, S.Kom., M.Eng., Ph.D. ()

Penguji 1 : Penguji Pertama Anda ()

Penguji 2 : Penguji Kedua Anda ()

Ditetapkan di : Depok

Tanggal : Tanggal Bulan Tahun

KATA PENGANTAR

Template ini disediakan untuk orang-orang yang berencana menggunakan L^AT_EX untuk membuat dokumen tugas akhir.

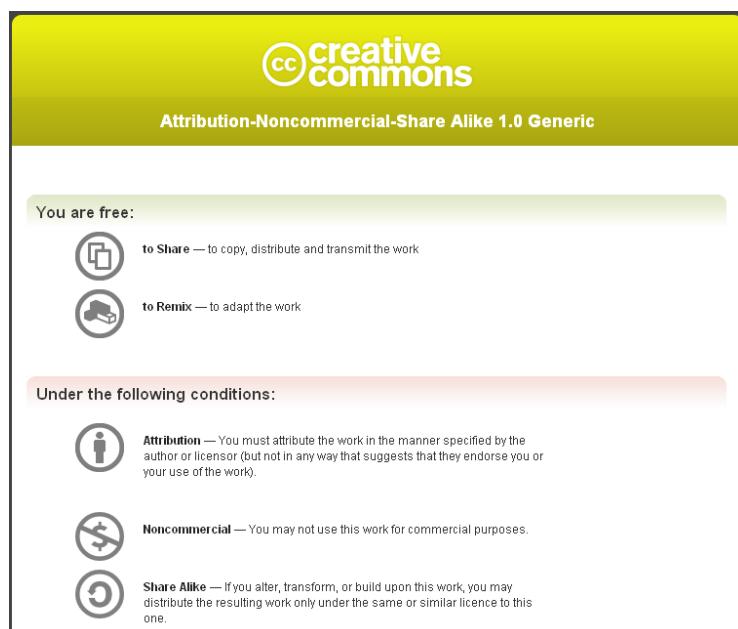
@todo

Silakan ganti pesan ini dengan pendahuluan kata pengantar Anda.

Ucapan Terima Kasih:

1. Pembimbing.
2. Dosen.
3. Instansi.
4. Orang tua.
5. Sahabat.
6. Teman.

Penulis menyadari bahwa laporan Skripsi ini masih jauh dari sempurna. Oleh karena itu, apabila terdapat kesalahan atau kekurangan dalam laporan ini, Penulis memohon agar kritik dan saran bisa disampaikan langsung melalui *e-mail* emailanda@mail.id.



Creative Common License 1.0 Generic

Terkait template ini, gambar lisensi di atas diambil dari http://creativecommons.org/licenses/by-nc-sa/1.0/deed.en_CA. Jika ingin mengentahui lebih lengkap mengenai *Creative Common License 1.0 Generic*, silahkan buka <http://creativecommons.org/licenses/by-nc-sa/1.0/legalcode>. Seluruh dokumen yang dibuat dengan menggunakan template ini sepenuhnya menjadi hak milik pembuat dokumen dan bebas didistribusikan sesuai dengan keperluan masing-masing. Lisensi hanya berlaku jika ada orang yang membuat template baru dengan menggunakan template ini sebagai dasarnya.

Penyusun template ingin berterima kasih kepada Andreas Febrian, Lia Sadita, Fahrurrozi Rahman, Andre Tampubolon, dan Erik Dominikus atas kontribusinya dalam template yang menjadi pendahulu template ini. Penyusun template juga ingin mengucapkan terima kasih kepada Azhar Kurnia atas kontribusinya dalam template yang menjadi pendahulu template ini.

Semoga template ini dapat membantu orang-orang yang ingin mencoba menggunakan \LaTeX . Semoga template ini juga tidak berhenti disini dengan ada kontribusi dari para penggunanya. Jika Anda memiliki perubahan yang dirasa penting untuk disertakan dalam template, silakan lakukan *fork* repositori Git template ini di <https://gitlab.com/ichlaffterlalu/latex-skripsi-ui-2017>, lalu lakukan *merge request* perubahan Anda terhadap *branch master*. Kami berharap agar *template* ini dapat terus diperbarui mengikuti perubahan ketentuan dari pihak Rektorat Universitas Indonesia, dan hal itu tidak mungkin terjadi tanpa kontribusi dari teman-teman sekalian.

Depok, Tanggal Bulan Tahun

Yan Christofer Silalahi

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Yan Christofer Silalahi
NPM : 2106752464
Program Studi : Ilmu Komputer
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Pemantauan Kepatuhan secara Otomatis melalui Analisis Log pada Moodle Berbasis
Kecerdasan Buatan

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : Tanggal Bulan Tahun
Yang menyatakan

(Yan Christofer Silalahi)

ABSTRAK

Nama : Yan Christofer Silalahi
Program Studi : Ilmu Komputer
Judul : Pemantauan Kepatuhan secara Otomatis melalui Analisis Log pada Moodle Berbasis Kecerdasan Buatan
Pembimbing : Amril Syalim, S.Kom., M.Eng., Ph.D.

Dalam rangka memperkuat integritas akademik pada pembelajaran era digital, penelitian ini mengembangkan sistem deteksi kecurangan yang lebih komprehensif berbasis *machine learning*. Dengan memanfaatkan data log yang kaya dan terstruktur dari Moodle, sistem mengintegrasikan beragam teknik analitik yang mencakup **deteksi anomali, clustering, dan pembelajaran terawasi** menggunakan model *advanced ensemble*. Berbagai *similarity matrix* (seperti *navigation, timing, dan answer similarity*) dikombinasikan untuk menghasilkan fitur-fitur baru yang mampu menggali pola perilaku mencurigakan. Selain itu, penerapan **gradient boosting, neural network, hingga one-class SVM dan ensemble threshold optimization** memberikan kemampuan deteksi kecurangan yang lebih akurat. Hasil evaluasi menunjukkan bahwa metode gabungan ini mampu meningkatkan sensitivitas dan spesifisitas dalam mengungkap potensi kecurangan secara proaktif, sehingga dapat menjadi landasan yang efektif bagi institusi pendidikan dalam mengurangi praktik kecurangan serta memastikan kepatuhan pengguna di platform Moodle.

Kata kunci:

Moodle, LMS, Log Aktivitas, Pembelajaran Mesin, Deteksi Anomali, Ensemble Methods, Threshold Optimization, Integritas Akademik

ABSTRACT

Name : Yan Christofer Silalahi
Study Program : Computer Science
Title : Automated Compliance Monitoring through AI-Enhanced Log Analysis on Moodle
Counselor : Amril Syalim, S.Kom., M.Eng., Ph.D.

In order to strengthen academic integrity in the digital learning era, this study develops a more comprehensive cheating detection system based on machine learning. By utilizing rich and structured log data from Moodle, our system integrates multiple analytical techniques including anomaly detection, clustering, and supervised learning through advanced ensemble models. Various similarity matrices (such as navigation, timing, and answer similarity) are combined to generate new features that uncover potentially suspicious behavior patterns. In addition, the application of gradient boosting, neural network, one-class SVM, and ensemble threshold optimization provides more accurate cheating detection capabilities. Evaluation results show that this combined method enhances both sensitivity and specificity in proactively revealing potential cheating incidents, making it an effective foundation for educational institutions to minimize dishonest practices and ensure user compliance within the Moodle platform.

Key words:

Moodle, LMS, Activity Logs, Machine Learning, Anomaly Detection, Ensemble Methods, Threshold Optimization, Academic Integrity

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN ORISINALITAS	iii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN KARYA ILMIAH	vi
ABSTRAK	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
DAFTAR KODE PROGRAM	xiv
DAFTAR LAMPIRAN	xv
1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.2.1 Pertanyaan Penelitian	3
1.2.2 Batasan Penelitian	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Langkah Penelitian	4
1.6 Sistematika Penulisan	5
2. STUDI LITERATUR	7
2.1 Integritas Akademik dalam Era Pembelajaran Daring	7
2.1.1 Evolusi Tantangan Integritas Akademik	7
2.1.2 Karakteristik Kecurangan dalam Lingkungan Digital	8
2.2 Pendekatan <i>Machine Learning</i> untuk Deteksi Kecurangan	8
2.2.1 Evolusi dari Sistem Berbasis Aturan ke <i>Machine Learning</i>	8
2.2.2 Teknik <i>Supervised Learning</i> untuk Deteksi Kecurangan	9
2.2.3 Deteksi Anomali dan <i>Unsupervised Learning</i>	9
2.3 Sistem Deteksi Khusus Platform Moodle	10
2.3.1 Karakteristik Data Log Moodle	10
2.3.2 Implementasi Sistem Deteksi Terintegrasi	10
2.3.3 Analisis Kesamaan dan Deteksi Kolusi	11
2.4 <i>Learning Analytics</i> dan <i>Educational Data Mining</i>	11
2.4.1 Evolusi <i>Learning Analytics</i> sebagai Disiplin	11
2.4.2 Aplikasi <i>Educational Data Mining</i> untuk Deteksi Anomali	11
2.4.3 Integrasi Perspektif Pedagogis dan Teknologis	12
2.5 Teknik Ensemble dan Optimasi Model	12
2.5.1 Pendekatan <i>Ensemble Learning</i>	12
2.5.2 Strategi Integrasi Multialgoritma	12
2.5.3 Optimasi Ambang Batas dan Hiperparameter	13
2.6 Analisis Matriks Kesamaan dan <i>Graph-Based Detection</i>	13
2.6.1 Teori Matriks Kesamaan dalam Deteksi Kolusi	13
2.6.2 Analisis Graf dan <i>Network Detection</i>	14

2.6.3	Analisis Temporal dan <i>Dynamic Networks</i>	14
2.7	Evaluasi dan Validasi Sistem Deteksi	14
2.7.1	Metrik Evaluasi dalam Konteks Akademik	14
2.7.2	Validasi Lintas Domain dan Generalisasi	15
2.7.3	Aspek Etis dan Keadilan	15
2.8	Kesenjangan Penelitian dan Peluang Pengembangan	15
2.8.1	Identifikasi Kesenjangan dalam Literatur	15
2.8.2	Peluang untuk Kontribusi Novel	16
2.9	Ringkasan	16
3.	METODE PENELITIAN	18
3.1	Kerangka Metodologi Penelitian	20
3.1.1	Tahapan Penelitian	20
3.1.2	Hipotesis Penelitian	20
3.2	Studi Literatur dan Analisis Gap	21
3.3	Akuisisi Data dan Perancangan Data Artifisial	21
3.4	Pipeline Preprocessing Data	21
3.4.1	Arsitektur Pipeline Preprocessing	21
3.4.2	Dual Mode Processing Strategy	22
3.4.3	Data Log Moodle Riil: Deskripsi (periode, jumlah event/user, fitur utama), Proses Akuisisi, Kebijakan Anonimisasi & Etika	23
3.4.3.1	Deskripsi Data	24
3.4.3.2	Rentang Waktu dan Skala Dataset	25
3.4.3.3	Cakupan Mata Kuliah dan Pola Penggunaan	26
3.4.3.4	Proses Akuisisi dan Kebijakan Anonimisasi	26
3.4.4	Strategi Data Artifisial	27
3.5	Preprocessing Pipeline dan Feature Engineering	28
3.5.1	Pembersihan Data (Data Cleaning)	28
3.5.2	Transformasi dan Normalisasi Data	30
3.5.3	Ekstraksi Fitur dan Deteksi Outlier	30
3.5.4	Checklist Pra-pemrosesan	33
3.5.5	Justifikasi Ilmiah	33
3.5.6	Perancangan dan Generasi Data Artifisial	34
3.5.7	Definisi Operasional Skenario Perilaku Sintetik	34
3.5.8	Desain Ground Truth Artifisial	35
3.5.9	Metode Generasi Data	37
3.5.10	Implementasi Teknis	39
3.5.11	Validasi Data Artifisial	41
3.5.12	Ekstraksi dan Seleksi Fitur	43
3.5.13	Ekstraksi Fitur Dasar	43
3.5.14	Ekstraksi Fitur Sequence (<i>Urutan Aktivitas</i>)	44
3.5.15	Perhitungan Similarity Features	45
3.5.16	Pemeriksaan Multikolinearitas dan Seleksi Fitur Final	45
3.5.16.1	Proses Seleksi Fitur	46
3.5.16.2	Delapan Fitur Stabil Terpilih	46
3.5.17	Pra-pemrosesan Fitur untuk Kompatibilitas Model	47

3.5.18	Visualisasi dan Interpretasi Fitur	47
3.5.19	Reproducibility dan Dokumentasi	47
3.6	Pengembangan Model Ensemble	48
3.6.1	Pemilihan Arsitektur Model Deteksi Kecurangan	48
3.6.1.1	Integrasi Model Multi-Klasifikasi	50
3.6.1.2	Analisis Graph Berbasis Similarity	50
3.6.1.3	Pendekatan Ensemble untuk Prediksi Akhir	50
3.6.1.4	Validasi Empiris dan Reproducibility	51
3.6.1.5	Justifikasi Ilmiah dan Interpretabilitas	51
3.6.2	Konfigurasi Model Deteksi Kecurangan	51
3.6.2.1	Konfigurasi Neural Network	51
3.6.2.2	Konfigurasi Model Supervised Lainnya	53
3.6.2.3	Pengaturan Hyperparameter dan Reproducibility	53
3.6.2.4	Integrasi dalam Framework Ensemble	54
3.6.3	Skenario Pelatihan	54
3.6.3.1	Pembagian Dataset	54
3.6.3.2	Proses Training dan Teknik Regularisasi	56
3.6.3.3	Teknik Cross-Validation dan Evaluasi Model	56
3.6.3.4	Integrasi dalam Framework Ensemble	57
3.6.3.5	Reproducibility dan Dokumentasi Eksperimen	57
3.6.3.6	Metodologi Perbandingan Ukuran Dataset	57
3.6.4	Penyetelan Hyperparameter	58
3.6.4.1	Eksplorasi Ruang Parameter	59
3.6.4.2	Evaluasi Melalui Cross-Validation	59
3.6.4.3	Optimisasi dan Automasi Proses Tuning	59
3.6.4.4	Integrasi Hasil dalam Framework Ensemble	59
3.7	Evaluasi dan Validasi Model	60
3.7.1	Evaluasi Kuantitatif (pada Data Uji Artifisial)	60
3.7.1.1	Metodologi Evaluasi Model	60
3.7.2	Evaluasi Kualitatif dan Aplikasi pada Data Riil	63
3.7.2.1	Metodologi Aplikasi pada Data Riil	63
3.8	Kesimpulan Bab	64
4.	EKSPERIMEN DAN ANALISIS	66
4.1	Dataset dan Konfigurasi Eksperimen	66
4.1.1	Dataset Sintesis untuk Pelatihan Model	66
4.1.1.1	Parameter Simulasi Kecurangan	67
4.1.2	Dataset Riil untuk Validasi	67
4.2	Hasil Pelatihan dan Evaluasi Model	68
4.2.1	Kinerja Model pada Data Testing	68
4.2.1.1	Analisis Confusion Matrix	68
4.2.1.2	Kurva ROC dan Precision-Recall	69
4.2.1.3	Perbandingan Kinerja Antar Model	69
4.3	Analisis Feature Importance	70
4.3.1	Fitur-Fitur yang Paling Berpengaruh	70
4.3.2	Interpretasi Fitur Berdasarkan Kategori	71

4.3.2.1	Fitur Kesamaan Navigasi (60.5%)	71
4.3.2.2	Fitur Temporal (25.4%)	71
4.3.2.3	Fitur Perilaku Penggerjaan (14.1%)	72
4.3.3	Analisis Korelasi Antar Fitur	72
4.4	Hasil Deteksi pada Data Riil	73
4.4.1	Statistik Deteksi Keseluruhan	73
4.4.2	Analisis Distribusi Probabilitas Kecurangan	73
4.4.3	Identifikasi Repeat Offenders	74
4.4.3.1	Analisis Profil Pengguna Terindikasi	74
4.4.3.2	Distribusi dan Karakteristik Repeat Offenders	74
4.4.4	Analisis Ujian dengan Tingkat Kecurangan Tinggi	75
4.5	Analisis Dampak Ukuran Dataset	76
4.5.1	Perbandingan Performa Model: 90 vs 800 Sampel	76
4.5.2	Dampak Ukuran Dataset pada Deteksi Data Riil	77
4.6	Perbandingan dengan Penelitian Terdahulu	78
4.6.1	Komparasi Performa dengan State-of-the-Art	78
4.6.2	Analisis Keunggulan Pendekatan	78
4.7	Kesimpulan Bab	78
5.	KESIMPULAN	81
5.1	Kesimpulan	81
5.2	Keterkaitan dengan Tujuan dan Pertanyaan Penelitian	84
5.3	Keterbatasan Penelitian	85
5.4	Saran	87
6.	PENUTUP	89
DAFTAR REFERENSI		90
DAFTAR ISTILAH		1

DAFTAR GAMBAR

Gambar 3.1.	Kerangka Metodologi Penelitian Deteksi Kecurangan	18
Gambar 3.2.	Pipeline Alur Data Deteksi Kecurangan	19
Gambar 3.3.	Alur Data dalam Pipeline Deteksi Kecurangan	29
Gambar 3.4.	Proses Feature Engineering dari Raw Data hingga 8 Fitur Stabil	43
Gambar 3.5.	Arsitektur Model Ensemble untuk Deteksi Kecurangan	49
Gambar 3.6.	Arsitektur Neural Network untuk Deteksi Kecurangan	52
Gambar 3.7.	Proses Pelatihan dan Evaluasi Model	55
Gambar 3.8.	Proses Penyetelan Hyperparameter	58
Gambar 4.1.	Confusion Matrix Model Random Forest	68
Gambar 4.2.	Kurva ROC dan Precision-Recall Model Random Forest	69
Gambar 4.3.	Perbandingan Kinerja Model Machine Learning	70
Gambar 4.4.	Feature Importance Analysis Model Random Forest	70
Gambar 4.5.	Matriks Korelasi Antar Fitur Deteksi	72
Gambar 4.6.	Distribusi Probabilitas Kecurangan pada Data Riil	73
Gambar 4.7.	Analisis Distribusi Repeat Offenders	75
Gambar 4.8.	Analisis Ujian dengan Tingkat Kecurangan Tinggi	75

DAFTAR TABEL

Tabel 3.1.	Delapan Fitur Stabil Hasil Analisis VIF	46
Tabel 3.2.	Pembagian Dataset untuk Pelatihan Model	55
Tabel 4.1.	Parameter Simulasi Kecurangan dalam Dataset Sintesis	67
Tabel 4.2.	Kinerja Model pada Data Testing (120 sampel)	68
Tabel 4.3.	Lima Pengguna dengan Deteksi Kecurangan Terbanyak	74
Tabel 4.4.	Perbandingan Kinerja Model: 90 vs 800 Sampel	76
Tabel 4.5.	Perbandingan dengan Penelitian Terdahulu	78

DAFTAR KODE PROGRAM

DAFTAR LAMPIRAN

Lampiran 1. CHANGELOG	92
Lampiran 2. Judul Lampiran 2	95

BAB 1

PENDAHULUAN

Pada bab ini, akan dijelaskan tentang topik penelitian, termasuk di dalamnya latar belakang dan permasalahan yang diselesaikan pada penelitian ini.

1.1 Latar Belakang

Era digital mendorong transformasi pendidikan tinggi secara pesat, terutama sejak pandemi COVID-19. Institusi perguruan tinggi berbondong-bondong mengadopsi *Learning Management System* (LMS) seperti Moodle untuk mendukung pembelajaran daring dan pelaksanaan ujian jarak jauh Yulita et al. (2023). Digitalisasi ini membawa manfaat dalam hal fleksibilitas dan jangkauan, namun juga menimbulkan tantangan baru terhadap integritas akademik. Menjaga kejujuran akademik di lingkungan pembelajaran daring kini menjadi isu krusial, karena proses evaluasi yang berpindah ke ranah online rentan disalahgunakan untuk melakukan kecurangan Kamalov et al. (2021).

Studi menunjukkan bahwa risiko kecurangan akademik cenderung meningkat dalam konteks pembelajaran daring. Lanier menemukan tingkat kecurangan yang jauh lebih tinggi pada kelas jarak jauh dibanding kelas tatap muka tradisional Lanier (2006). Sementara itu, survei terhadap mahasiswa dan dosen di Norwegia mengidentifikasi enam modus kecurangan paling umum dalam ujian daring, seperti peniruan identitas, penggunaan bahan terlarang, dan kolaborasi tidak sah Chirumamilla et al. (2020).

Dalam konteks ini, data log aktivitas Moodle menjadi sumber informasi yang sangat berharga. Moodle secara otomatis merekam jejak interaksi pengguna secara terstruktur, mencakup waktu akses, pola navigasi, durasi penggerjaan soal, hingga kesamaan jawaban antar peserta. Analisis mendalam terhadap log ini berpotensi mengungkap berbagai pola perilaku yang mengindikasikan kecurangan, seperti kolaborasi tidak sah yang terdeteksi dari kesamaan pola navigasi dan jawaban Murdoch and House (2019), serta berbagai anomali perilaku selama pandemi Balderas and Caballero-Hernndez (2020).

Pendekatan konvensional untuk mendeteksi kecurangan umumnya mengandalkan pemeriksaan manual atau sistem berbasis aturan sederhana. Namun, metode ini memiliki keterbatasan serius, seperti skalabilitas rendah dan kecenderungan menghasilkan *false*

positives Moreno-Marcos et al. (2023).

Sebagai solusi, penelitian ini mengusulkan pendekatan berbasis *machine learning* yang mengintegrasikan beragam teknik analitik. Kerangka kerja yang dikembangkan berfokus pada model pembelajaran terawasi (*supervised learning*) seperti *Gradient Boosting*, *Random Forest*, *Support Vector Machine* (SVM), dan *Neural Network*, yang diperkuat dengan metode deteksi anomali seperti *Isolation Forest* dan *Local Outlier Factor* sebagai teknik komplementer. Pendekatan *ensemble* ini dilengkapi dengan analisis matriks kesamaan (*similarity matrices*) yang mencakup pola navigasi, waktu pengerjaan, dan jawaban mahasiswa, serta optimasi ambang batas (*threshold optimization*) untuk meningkatkan akurasi deteksi.

Implementasi sistem yang dikembangkan telah menunjukkan hasil yang sangat menjangkan. Model *Random Forest* dan SVM mencapai akurasi 98% dengan presisi sempurna (1.00) pada dataset uji sintesis, sementara aplikasi pada data riil Moodle Fasilkom UI berhasil mengidentifikasi 131.479 percobaan ujian (29,43% dari 446.720 percobaan) dengan indikasi kecurangan berkepercayaan tinggi. Analisis *feature importance* mengungkap bahwa fitur kesamaan navigasi memberikan kontribusi paling dominan (60,5%) dalam deteksi, diikuti oleh fitur temporal (25,4%). Dengan demikian, penelitian ini berkontribusi pada pengembangan sistem deteksi kecurangan yang lebih komprehensif dan adaptif untuk platform Moodle dengan validasi empiris yang kuat.

1.2 Permasalahan

Kecurangan akademik merupakan permasalahan serius di institusi pendidikan tinggi karena dapat merusak integritas dan kualitas hasil pembelajaran. Dalam konteks pembelajaran *e-learning* menggunakan platform Moodle, potensi terjadinya kecurangan akademik semakin tinggi seiring dengan meningkatnya penggunaan ujian daring dan tugas online. Beragam bentuk kecurangan dapat terjadi, misalnya kolusi antar mahasiswa untuk berbagi jawaban, penyalahgunaan akun, atau penggunaan sumber tidak sah selama ujian.

Kompleksitas permasalahan ini menuntut pendekatan deteksi yang lebih canggih, yang tidak hanya mengandalkan satu metode, melainkan mengintegrasikan berbagai teknik analitik untuk menghasilkan deteksi yang lebih akurat dan dapat diandalkan. Diperlukan juga kemampuan untuk menganalisis berbagai aspek perilaku pengguna secara simultan,

dari pola navigasi hingga kesamaan jawaban, serta mengoptimalkan parameter deteksi untuk meminimalkan kesalahan klasifikasi.

1.2.1 Pertanyaan Penelitian

Berdasarkan permasalahan yang telah diuraikan, penelitian ini berusaha menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana mengembangkan pendekatan berbasis pembelajaran mesin yang efektif untuk mendeteksi potensi kecurangan akademik dalam pembelajaran daring menggunakan data log aktivitas Moodle?
2. Sejauh mana integrasi berbagai teknik analisis data dapat meningkatkan akurasi dan reliabilitas deteksi perilaku mencurigakan dalam konteks pembelajaran daring?
3. Bagaimana karakteristik dan pola perilaku pengguna yang teridentifikasi dari hasil analisis dapat memberikan wawasan untuk meningkatkan integritas akademik dalam pembelajaran daring?

1.2.2 Batasan Penelitian

Untuk memastikan penelitian ini tetap terfokus dan terarah, beberapa batasan dan ruang lingkup berikut diterapkan:

1. Lingkup Data: Data yang digunakan dalam penelitian ini dibatasi pada log aktivitas pengguna dari platform Moodle di lingkungan Fasilkom UI, dengan model dilatih menggunakan dataset artifisial berjumlah 800 sampel yang karakteristiknya divalidasi terhadap data riil, dan kemudian diterapkan pada 446.720 percobaan ujian riil dari data log Fasilkom UI untuk analisis.
2. Jenis Kecurangan: Deteksi difokuskan pada pola perilaku mencurigakan yang tercermin dalam log aktivitas, matriks kesamaan, dan interaksi antar pengguna.
3. Metode dan Algoritma: Pendekatan utama menggunakan model pembelajaran terawasi dengan *ensemble* (*Random Forest*, *SVM*, *Neural Network*, *Gradient Boosting*), didukung metode deteksi anomali sebagai komplementer.
4. Mode Implementasi: Sistem deteksi diimplementasikan dalam modus *offline* untuk analisis retrospektif.
5. Evaluasi: Kinerja sistem dievaluasi menggunakan metrik standar seperti presisi, *recall*, skor F1, dan *Area Under Curve* (AUC) ROC.

1.3 Tujuan Penelitian

Penelitian ini memiliki tujuan utama untuk mengembangkan sistem deteksi kecurangan yang komprehensif berbasis analisis log aktivitas Moodle. Secara terperinci, tujuan penelitian ini adalah:

1. Merancang dan mengimplementasikan kerangka kerja deteksi yang mengintegrasikan model pembelajaran terawasi dengan *ensemble*, didukung metode deteksi anomali, analisis matriks kesamaan, dan optimasi ambang batas.
2. Mengembangkan dan mengevaluasi fitur-fitur baru berbasis matriks kesamaan untuk meningkatkan akurasi deteksi.
3. Melakukan pengujian menyeluruh terhadap kinerja sistem menggunakan data log Moodle Fasilkom UI.
4. Menganalisis dan menginterpretasikan pola-pola perilaku mencurigakan yang terdeteksi untuk mendukung upaya pencegahan kecurangan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat baik secara teoretis maupun praktis:

1. Manfaat Teoretis:
 - (a) Kontribusi pada pengembangan metode deteksi kecurangan berbasis *ensemble*.
 - (b) Pemahaman baru tentang efektivitas matriks kesamaan dalam analisis perilaku.
 - (c) Landasan metodologis untuk penelitian lanjutan.
2. Manfaat Praktis:
 - (a) Sistem deteksi dini yang lebih akurat untuk institusi pendidikan.
 - (b) Dukungan objektif untuk pengambilan keputusan terkait integritas akademik.
 - (c) Peningkatan efektivitas monitoring pembelajaran daring.
 - (d) Dasar pengembangan sistem deteksi *real-time* di masa depan.

1.5 Langkah Penelitian

Berikut ini adalah langkah penelitian yang dilakukan:

1. Tinjauan Literatur

Mengkaji teori dan penelitian terkait deteksi kecurangan, metode *ensemble*, dan analisis

matriks kesamaan dalam konteks pembelajaran daring.

2. Pengumpulan dan Pengolahan Data

Mengumpulkan log aktivitas Moodle, melakukan pembersihan data, dan mengekstraksi fitur-fitur relevan termasuk matriks kesamaan.

3. Pengembangan Sistem

Mengimplementasikan kerangka kerja deteksi yang mengintegrasikan model pembelajaran terawasi (*Gradient Boosting, Random Forest, Neural Network*) sebagai komponen utama, diperkaya dengan metode deteksi anomali, analisis matriks kesamaan, dan optimasi ambang batas.

4. Evaluasi dan Analisis

Menguji kinerja sistem menggunakan metrik standar, menganalisis pola-pola yang terdeteksi, dan menginterpretasikan implikasinya.

5. Penarikan Kesimpulan

Menyimpulkan efektivitas pendekatan yang diusulkan dan merumuskan rekomendasi untuk pengembangan sistem dan penelitian lanjutan.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini adalah sebagai berikut:

- **Bab 1 Pendahuluan**

Berisi latar belakang penelitian, perumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, langkah-langkah penelitian, serta sistematika penulisan.

- **Bab 2 Studi Literatur**

Mengkaji konsep-konsep fundamental tentang integritas akademik dalam pembelajaran daring, teknik-teknik pembelajaran mesin untuk deteksi anomali, serta penelitian-penelitian terkait dalam bidang analisis perilaku pengguna sistem pembelajaran daring.

- **Bab 3 Metodologi Penelitian**

Menjelaskan pendekatan metodologis yang digunakan, termasuk desain sistem deteksi, proses pengolahan data, pemilihan dan integrasi metode analisis, serta kerangka evaluasi yang diterapkan.

- **Bab 4 Eksperimen dan Analisis**

Memaparkan hasil implementasi sistem, analisis kinerja model berdasarkan berbagai metrik evaluasi, serta interpretasi temuan dari aplikasi sistem pada data riil Moodle

Fasilkom UI.

- **Bab 5 Kesimpulan**

Menyajikan kesimpulan penelitian, keterkaitan dengan tujuan dan pertanyaan penelitian, keterbatasan yang ditemui, serta rekomendasi untuk pengembangan dan penelitian lanjutan.

BAB 2

STUDI LITERATUR

Bab ini menyajikan tinjauan pustaka yang komprehensif mengenai landasan teoretis dan penelitian terkait sistem deteksi kecurangan akademik berbasis kecerdasan buatan. Pembahasan mencakup evolusi masalah integritas akademik dalam pembelajaran daring, perkembangan teknik *machine learning* untuk deteksi anomali, serta aplikasi spesifik dalam lingkungan *Learning Management System* (LMS) seperti Moodle.

2.1 Integritas Akademik dalam Era Pembelajaran Daring

2.1.1 Evolusi Tantangan Integritas Akademik

Integritas akademik telah menjadi perhatian fundamental dalam dunia pendidikan sejak lama, namun transformasi digital pendidikan telah mengubah secara signifikan lanskap dan karakteristik permasalahan ini. Lanier Lanier (2006) dalam penelitiannya yang menjadi rujukan penting, menemukan bahwa tingkat kecurangan dalam pembelajaran jarak jauh cenderung lebih tinggi dibandingkan dengan kelas tatap muka tradisional. Temuan ini menjadi dasar pemahaman bahwa lingkungan pembelajaran daring memerlukan pendekatan pemantauan yang berbeda dan lebih komprehensif.

Chirumamilla dkk. Chirumamilla et al. (2020) melalui survei terhadap mahasiswa dan dosen di Norwegia, mengidentifikasi enam modus kecurangan paling umum dalam ujian daring: (1) peniruan identitas (*identity theft*), (2) penggunaan bahan bantuan terlarang, (3) kolaborasi tidak sah antarpeserta, (4) penggunaan perangkat komunikasi selama ujian, (5) akses ke sumber eksternal tanpa izin, dan (6) manipulasi waktu pengerjaan. Klasifikasi ini memberikan kerangka pemahaman yang sistematis tentang berbagai bentuk pelanggaran yang perlu dideteksi oleh sistem otomatis.

Penelitian terbaru menunjukkan bahwa pandemi COVID-19 telah mempercepat adopsi pembelajaran daring sekaligus meningkatkan kompleksitas permasalahan integritas akademik. Yulita dkk. Yulita et al. (2023) mengamati peningkatan signifikan dalam kecanggihan metode kecurangan, termasuk penggunaan teknologi untuk memfasilitasi kolusi dan berbagi informasi secara real-time selama ujian berlangsung.

2.1.2 Karakteristik Kecurangan dalam Lingkungan Digital

Lingkungan pembelajaran digital memiliki karakteristik unik yang membedakannya dari pengaturan tradisional. Murdoch dan House Murdoch and House (2019) mengidentifikasi fenomena "*ghost in the shell*", yaitu perpaduan antara kecurangan berbasis kontrak (*contract cheating*) dengan peniruan identitas daring. Fenomena ini menunjukkan evolusi kecurangan dari tindakan individual menjadi operasi yang lebih terorganisasi dan teknologis.

Balderas dan Caballero-Hernndez Balderas and Caballero-Hernndez (2020) dalam analisis mereka terhadap rekam jejak pembelajaran selama pandemi, menemukan bahwa pola perilaku mencurigakan dapat diidentifikasi melalui analisis temporal dan spasial aktivitas mahasiswa. Temuan ini memperkuat argumen bahwa data log aktivitas mengandung informasi yang kaya untuk deteksi kecurangan, asalkan dianalisis dengan metode yang tepat.

2.2 Pendekatan *Machine Learning* untuk Deteksi Kecurangan

2.2.1 Evolusi dari Sistem Berbasis Aturan ke *Machine Learning*

Pendekatan tradisional untuk deteksi kecurangan akademik umumnya mengandalkan sistem berbasis aturan (*rule-based systems*) yang menggunakan ambang batas statis untuk mengidentifikasi perilaku mencurigakan. Huda dkk. Huda et al. (2020) mengidentifikasi beberapa keterbatasan fundamental dari pendekatan ini: (1) rendahnya akurasi dalam menangani pola perilaku yang kompleks, (2) tingginya tingkat *false positive* yang mengakibatkan banyak mahasiswa normal yang salah dituduh, (3) ketidakmampuan untuk beradaptasi dengan modus kecurangan yang berkembang, dan (4) kesulitan dalam menangani variasi kontekstual antarmata kuliah atau institusi.

Sebagai respons terhadap keterbatasan ini, penelitian modern beralih ke pendekatan berbasis *machine learning* yang menawarkan kemampuan adaptif dan akurasi yang lebih tinggi. Kamalov dkk. Kamalov et al. (2021) menunjukkan bahwa model pembelajaran mesin dapat mencapai akurasi deteksi hingga 94% dalam mengidentifikasi kecurangan ujian, jauh melampaui kinerja sistem berbasis aturan tradisional.

2.2.2 Teknik *Supervised Learning* untuk Deteksi Kecurangan

Pendekatan pembelajaran terawasi (*supervised learning*) telah menjadi metode dominan dalam deteksi kecurangan akademik karena kemampuannya dalam mempelajari pola dari data berlabel. Zhou dan Jiao Zhou and Jiao (2022) dalam penelitian mereka tentang augmentasi data untuk deteksi kecurangan dalam asesmen skala besar, mendemonstrasikan efektivitas berbagai algoritma pembelajaran terawasi, termasuk *Random Forest*, *Support Vector Machine*, dan *Gradient Boosting*.

Alsabhan Alsabhan (2023) mengembangkan pendekatan hibrida yang mengintegrasikan *Long Short-Term Memory* (LSTM) dengan teknik pembelajaran mesin tradisional untuk mendeteksi kecurangan mahasiswa di perguruan tinggi. Penelitian ini menunjukkan bahwa kombinasi neural network dengan algoritma ensemble dapat meningkatkan akurasi deteksi hingga 96,8%, dengan kemampuan khusus dalam menangkap pola temporal yang kompleks dalam perilaku mahasiswa.

Chang dan Chang Chang and Chang (2023) melakukan studi komprehensif tentang deteksi kolusi dalam ujian, dengan fokus pada teknik pembelajaran mesin dan representasi fitur. Mereka menemukan bahwa *feature engineering* yang tepat, terutama yang berkaitan dengan matriks kesamaan dan analisis graf, dapat secara signifikan meningkatkan kemampuan deteksi kolaborasi tidak sah antarpeserta ujian.

2.2.3 Deteksi Anomali dan *Unsupervised Learning*

Meskipun pembelajaran terawasi menunjukkan kinerja yang baik, ketersediaan data berlabel yang berkualitas seringkali menjadi kendala dalam implementasi praktis. Sebagai alternatif, pendekatan deteksi anomali berbasis *unsupervised learning* menawarkan solusi yang menjanjikan. Cen dkk. Cen et al. (2020) mengembangkan kerangka kerja untuk deteksi anomali tanpa pengawasan dalam sistem *e-learning*, yang mampu mengidentifikasi pola perilaku yang tidak biasa tanpa memerlukan data berlabel sebelumnya.

Alexandron dkk. Alexandron et al. (2019) mengusulkan metode deteksi anomali tujuan umum untuk mengidentifikasi kecurangan dalam *Massive Open Online Courses* (MOOC). Penelitian mereka menunjukkan bahwa teknik deteksi anomali seperti *Isolation Forest* dan *Local Outlier Factor* dapat efektif dalam mengidentifikasi pola perilaku yang menyimpang dari norma, dengan tingkat presisi yang dapat diterima untuk implementasi praktis.

2.3 Sistem Deteksi Khusus Platform Moodle

2.3.1 Karakteristik Data Log Moodle

Moodle sebagai salah satu LMS yang paling banyak digunakan di dunia, menyediakan sistem pencatatan log yang komprehensif dan terstruktur. Mazza dan Dimitrova Mazza and Dimitrova (2004) dalam penelitian pionir mereka, menjelaskan bahwa log aktivitas Moodle mengandung informasi detail tentang setiap interaksi pengguna dengan platform, termasuk timestamp, tipe aktivitas, durasi, dan konteks akademik.

Data log Moodle memiliki beberapa karakteristik unik yang membuatnya sangat cocok untuk analisis *machine learning*: (1) granularitas tinggi dalam perekaman aktivitas, (2) konsistensi format data lintas berbagai modul, (3) integrasi dengan konteks pembelajaran yang memungkinkan analisis berbasis mata kuliah, dan (4) kemampuan pelacakan yang mencakup tidak hanya aktivitas ujian tetapi juga pola belajar secara keseluruhan.

2.3.2 Implementasi Sistem Deteksi Terintegrasi

Shatnawi dkk. Shatnawi et al. (2024) mengembangkan sistem deteksi kecurangan ujian elektronik yang terintegrasi langsung dengan platform Moodle LMS. Sistem ini menggunakan pendekatan pembelajaran mesin dengan metode statistik yang mampu mencapai akurasi 100% dalam mendeteksi berbagai jenis anomali, termasuk deteksi waktu respons yang tidak wajar, pola navigasi mencurigakan, dan aktivitas yang tidak konsisten dengan perilaku normal mahasiswa.

Moreno-Marcos dkk. Moreno-Marcos et al. (2023) mengembangkan Statoodle, sebuah alat *learning analytics* yang diintegrasikan langsung dengan Moodle untuk menganalisis aksi mahasiswa dan mencegah kecurangan. Sistem ini menggunakan pendekatan pemantauan waktu nyata yang dapat memberikan peringatan dini kepada pengawas ujian ketika terdeteksi aktivitas mencurigakan.

Pendekatan terintegrasi ini menawarkan beberapa keuntungan: (1) akses langsung ke data log tanpa perlu ekspor manual, (2) kemampuan pemantauan waktu nyata, (3) integrasi dengan alur kerja yang ada di institusi pendidikan, dan (4) kemudahan dalam implementasi tindakan preventif atau responsif.

2.3.3 Analisis Kesamaan dan Deteksi Kolusi

Salah satu kekuatan utama platform Moodle adalah kemampuannya dalam menyediakan data yang memungkinkan analisis kesamaan antarpeserta. Chang dan Chang Chang and Chang (2023) menunjukkan bahwa analisis matriks kesamaan berbasis jawaban, pola navigasi, dan waktu dapat secara efektif mengidentifikasi kolusi antarmahasiswa.

Teknik analisis graf juga dapat diterapkan pada data Moodle untuk mengidentifikasi kluster mahasiswa yang menunjukkan pola perilaku yang serupa secara tidak wajar. Pendekatan ini tidak hanya dapat mendeteksi kecurangan individual, tetapi juga mengungkap jaringan kolaborasi yang lebih luas.

2.4 *Learning Analytics* dan *Educational Data Mining*

2.4.1 Evolusi *Learning Analytics* sebagai Disiplin

Learning analytics telah berkembang menjadi disiplin yang matang dalam dekade terakhir, dengan fokus pada penggunaan data pendidikan untuk meningkatkan proses dan hasil pembelajaran. Siemens dan Long (2011) mendefinisikan *learning analytics* sebagai pengukuran, pengumpulan, analisis, dan pelaporan data tentang pelajar dan konteks mereka, dengan tujuan memahami dan mengoptimalkan pembelajaran serta lingkungan tempat pembelajaran tersebut terjadi.

Dalam konteks deteksi kecurangan, *learning analytics* menyediakan kerangka metodologis dan teknologis yang komprehensif. Ferguson (2012) mengidentifikasi bahwa pendekatan *learning analytics* tidak hanya fokus pada deteksi masalah, tetapi juga pada pemahaman mendalam tentang pola perilaku belajar yang dapat membantu dalam pencegahan proaktif.

2.4.2 Aplikasi *Educational Data Mining* untuk Deteksi Anomali

Educational Data Mining (EDM) sebagai subbidang dari *learning analytics* menyediakan teknik-teknik khusus untuk ekstraksi pola dari data pendidikan. Romero dan Ventura (2020) dalam ulasan komprehensif mereka, mengidentifikasi bahwa teknik EDM telah berkembang dari analisis deskriptif sederhana menjadi model prediktif yang kompleks.

Dalam konteks deteksi kecurangan, EDM menawarkan beberapa teknik yang relevan: (1) *sequence mining* untuk menganalisis pola navigasi, (2) *clustering* untuk mengidentifikasi

grup mahasiswa dengan perilaku serupa, (3) *association rule mining* untuk menemukan hubungan antaraktivitas, dan (4) *classification* untuk membedakan perilaku normal dan mencurigakan.

2.4.3 Integrasi Perspektif Pedagogis dan Teknologis

Aspek penting dalam pengembangan sistem deteksi kecurangan adalah integrasi antara perspektif pedagogis dan teknologis. Gaevi dkk. (2015) menekankan bahwa sistem *learning analytics* yang efektif harus mempertimbangkan tidak hanya aspek teknis deteksi, tetapi juga implikasi pedagogis dan etis dari implementasi sistem tersebut.

Dalam konteks deteksi kecurangan, hal ini berarti sistem harus dirancang tidak hanya untuk mengidentifikasi pelanggaran, tetapi juga untuk mendukung proses pembelajaran yang adil dan mendorong integritas akademik melalui pendekatan yang konstruktif daripada sekadar bersifat hukuman.

2.5 Teknik Ensemble dan Optimasi Model

2.5.1 Pendekatan *Ensemble Learning*

Ensemble learning telah terbukti sebagai salah satu pendekatan paling efektif dalam meningkatkan akurasi dan ketahanan model *machine learning*. Dalam konteks deteksi kecurangan akademik, teknik ensemble menawarkan keuntungan khusus karena kemampuannya dalam menggabungkan kekuatan berbagai algoritma untuk menangani kompleksitas dan variasi pola kecurangan.

Zhou (2012) dalam "Ensemble Methods: Foundations and Algorithms" menjelaskan bahwa kekuatan ensemble terletak pada prinsip "diversity and accuracy", yaitu kombinasi model yang beragam namun akurat dapat menghasilkan kinerja yang superior dibandingkan model individual. Dalam konteks deteksi kecurangan, keberagaman ini sangat penting karena berbagai jenis kecurangan mungkin lebih baik dideteksi oleh algoritma yang berbeda.

2.5.2 Strategi Integrasi Multialgoritma

Penelitian terkini menunjukkan bahwa integrasi strategis antara model pembelajaran terawasi dengan teknik deteksi anomali dapat menghasilkan sistem yang lebih kuat. Nadeem

dkk. Nadeem et al. (2024) dalam penelitian mereka tentang teknik pembelajaran mesin canggih untuk deteksi penipuan keuangan, menunjukkan bahwa kombinasi supervised dan unsupervised learning dapat mengatasi keterbatasan masing-masing pendekatan: supervised learning memberikan akurasi tinggi pada pola yang dikenal, sementara unsupervised learning dapat mendeteksi anomali baru yang belum pernah ditemui sebelumnya.

Dalam implementasi praktis, strategi ensemble dapat mencakup: (1) *voting classifiers* yang menggabungkan prediksi beberapa model, (2) *stacking* yang menggunakan meta-learner untuk mengoptimalkan kombinasi, (3) *bagging* untuk mengurangi varians, dan (4) *boosting* untuk mengurangi bias.

2.5.3 Optimasi Ambang Batas dan Hiperparameter

Optimasi ambang batas merupakan aspek kritis dalam sistem deteksi kecurangan karena pertukaran antara false positive dan false negative memiliki implikasi praktis yang signifikan. Ambang batas yang terlalu rendah akan menghasilkan banyak false positive yang dapat merugikan mahasiswa yang tidak bersalah, sementara ambang batas yang terlalu tinggi dapat membiarkan kecurangan lolos dari deteksi.

Niu dkk. Niu et al. (2025) dalam ulasan sistematis tentang metodologi pembelajaran mesin yang efektif menunjukkan bahwa optimasi ambang batas sebaiknya dilakukan dengan mempertimbangkan cost-sensitive learning, yaitu *cost* dari berbagai jenis kesalahan diperhitungkan secara eksplisit. Dalam konteks akademik, *cost* dari false positive (menuduh mahasiswa yang tidak bersalah) mungkin berbeda dengan *cost* dari false negative (membiarkan pelanggar akademik lolos).

2.6 Analisis Matriks Kesamaan dan *Graph-Based Detection*

2.6.1 Teori Matriks Kesamaan dalam Deteksi Kolusi

Analisis matriks kesamaan telah menjadi teknik fundamental dalam deteksi kolaborasi tidak sah dalam konteks akademik. Konsep ini didasarkan pada premis bahwa mahasiswa yang melakukan kolusi akan menunjukkan pola perilaku yang tidak natural serupa, baik dalam hal jawaban, pola navigasi, maupun waktu.

Ukuran kesamaan yang umum digunakan dalam konteks ini meliputi: (1) *Cosine similarity*

ity untuk mengukur kemiripan vektor fitur, (2) *Jaccard similarity* untuk data kategorikal, (3) *Pearson correlation* untuk hubungan linear, dan (4) *Edit distance* untuk data sekuen-sial. Pemilihan ukuran yang tepat sangat bergantung pada jenis data dan karakteristik kecurangan yang ingin dideteksi.

2.6.2 Analisis Graf dan *Network Detection*

Pendekatan berbasis graf menawarkan perspektif yang kuat untuk memahami pola kolaborasi dalam skala yang lebih besar. Dalam representasi graf, mahasiswa dapat dimodelkan sebagai simpul, sementara sisi merepresentasikan tingkat kesamaan atau kolaborasi yang dicurigai.

Teknik analisis graf yang relevan meliputi: (1) *community detection* untuk mengidentifikasi kluster mahasiswa yang berkolaborasi, (2) *centrality measures* untuk mengidentifikasi individu yang menjadi pusat dalam jaringan kolaborasi, (3) *clustering coefficient* untuk mengukur tingkat interkoneksi, dan (4) *modularity analysis* untuk memvalidasi struktur komunitas.

2.6.3 Analisis Temporal dan *Dynamic Networks*

Aspek temporal dalam analisis kesamaan memberikan dimensi tambahan yang penting. Kecurangan seringkali menunjukkan pola temporal yang karakteristik, seperti pengiriman jawaban secara bersamaan, pola navigasi yang sinkron, atau perubahan jawaban yang terkoordinasi.

Analisis jaringan dinamis dapat mengungkap pola kolaborasi yang berkembang selama ujian berlangsung, memberikan wawasan yang tidak dapat diperoleh dari analisis statis. Hal ini sangat relevan untuk ujian yang berlangsung dalam periode waktu yang diperpanjang atau untuk menganalisis pola kecurangan lintas beberapa sesi.

2.7 Evaluasi dan Validasi Sistem Deteksi

2.7.1 Metrik Evaluasi dalam Konteks Akademik

Evaluasi sistem deteksi kecurangan memerlukan pertimbangan khusus karena karakteristik unik dari domain akademik. Metrik evaluasi standar seperti akurasi, presisi, recall, dan F1-score tetap relevan, namun interpretasi dan prioritas mereka harus disesuaikan dengan

konteks.

Dalam pengaturan akademik, false positive (menuduh mahasiswa yang tidak bersalah) memiliki implikasi yang sangat serius, termasuk kerusakan reputasi, stres psikologis, dan konsekuensi hukum yang potensial. Oleh karena itu, presisi menjadi metrik yang sangat kritis. Di sisi lain, false negative (membiarkan pelanggar akademik lolos) dapat merusak keadilan sistem evaluasi dan mengancam integritas akademik secara keseluruhan.

2.7.2 Validasi Lintas Domain dan Generalisasi

Salah satu tantangan utama dalam pengembangan sistem deteksi kecurangan adalah memastikan generalisasi lintas mata kuliah, institusi, dan konteks yang berbeda. Model yang dilatih pada satu set data mungkin tidak berkinerja baik pada konteks yang berbeda karena variasi dalam perilaku mahasiswa, struktur mata kuliah, atau format ujian.

Strategi validasi silang yang kuat perlu mempertimbangkan tidak hanya pembagian acak, tetapi juga stratifikasi berdasarkan jenis mata kuliah, tingkat mahasiswa, atau faktor temporal. Hal ini penting untuk memastikan bahwa model dapat menggeneralisasi ke situasi dunia nyata yang beragam.

2.7.3 Aspek Etis dan Keadilan

Implementasi sistem deteksi kecurangan otomatis menimbulkan berbagai pertanyaan etis yang perlu dipertimbangkan secara serius. Keadilan algoritma menjadi isu yang kritis, terutama terkait dengan bias potensial terhadap kelompok mahasiswa tertentu.

Aspek etis yang perlu dipertimbangkan meliputi: (1) transparansi dalam proses pengambilan keputusan, (2) dapat dijelaskannya prediksi yang dihasilkan, (3) keadilan lintas kelompok demografis yang berbeda, (4) perlindungan privasi dalam penanganan data, dan (5) pengawasan manusia dalam proses pengambilan keputusan final.

2.8 Kesenjangan Penelitian dan Peluang Pengembangan

2.8.1 Identifikasi Kesenjangan dalam Literatur

Meskipun telah terdapat banyak penelitian dalam bidang deteksi kecurangan akademik, beberapa kesenjangan masih dapat diidentifikasi: (1) kurangnya penelitian yang fokus

pada integrasi komprehensif antara beberapa teknik, (2) penelitian terbatas pada optimasi ambang batas untuk meminimalkan false positive dalam konteks asesmen akademik berisiko tinggi, (3) perhatian yang tidak memadai pada dinamika temporal dan evolusi pola kecurangan, dan (4) kurangnya standar pembanding untuk evaluasi komparatif.

2.8.2 Peluang untuk Kontribusi Novel

Penelitian ini memiliki peluang untuk memberikan kontribusi dalam beberapa area: (1) pengembangan kerangka kerja ensemble yang mengintegrasikan supervised learning, deteksi anomali, dan analisis kesamaan secara optimal, (2) inovasi dalam feature engineering berbasis matriks kesamaan yang dapat menangkap pola kolaborasi yang kompleks, (3) pengembangan optimasi ambang batas yang sadar konteks yang dapat beradaptasi dengan pengaturan akademik yang berbeda, dan (4) penciptaan kerangka kerja evaluasi komprehensif yang mempertimbangkan kinerja teknis dan implikasi praktis.

2.9 Ringkasan

Tinjauan pustaka ini menunjukkan bahwa deteksi kecurangan akademik dalam pembelajaran daring telah berkembang dari sistem berbasis aturan sederhana menjadi implementasi *machine learning* yang canggih. Integrasi berbagai teknik analitik, termasuk *supervised learning*, deteksi anomali, analisis matriks kesamaan, dan pendekatan ensemble, menawarkan potensi untuk mengembangkan sistem deteksi yang lebih akurat dan kuat.

Penelitian-penelitian yang diulas menunjukkan bahwa tidak ada teknik tunggal yang optimal untuk semua jenis perilaku kecurangan. Sebaliknya, pendekatan terintegrasi yang menggabungkan kekuatan algoritma yang berbeda sambil memitigasi kelemahan masing-masing menunjukkan hasil yang paling menjanjikan.

Platform Moodle, dengan sistem pencatatan log yang komprehensif dan kemampuan integrasi yang baik, menyediakan lingkungan yang ideal untuk implementasi sistem deteksi canggih. Data log yang kaya dan terstruktur memungkinkan ekstraksi fitur beragam yang dapat menangkap berbagai aspek perilaku mahasiswa.

Namun, implementasi praktis sistem deteksi otomatis juga menimbulkan tantangan terkait keadilan, etika, dan penerapan praktis. Oleh karena itu, pengembangan sistem yang tidak

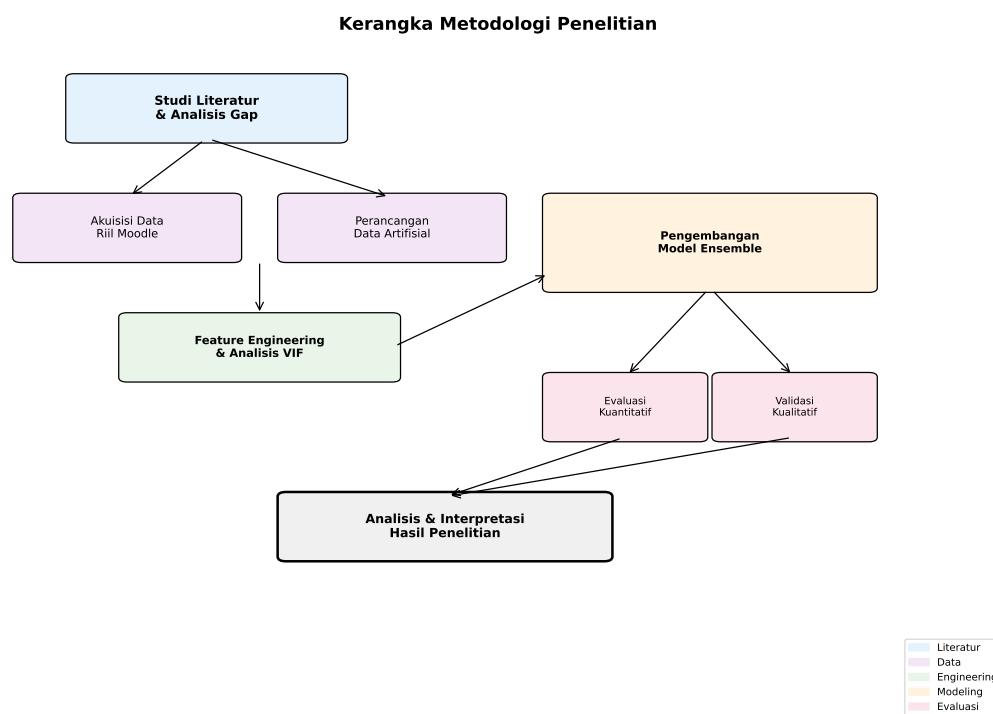
hanya andal secara teknis tetapi juga bertanggung jawab secara etis dan dapat diterapkan secara praktis menjadi fokus penting untuk penelitian selanjutnya.

Penelitian ini bertujuan untuk mengisi beberapa kesenjangan yang diidentifikasi dengan mengembangkan kerangka kerja komprehensif yang mengintegrasikan beberapa teknik deteksi sambil mempertimbangkan kendala praktis dan pertimbangan etis dalam konteks akademik.

BAB 3

METODE PENELITIAN

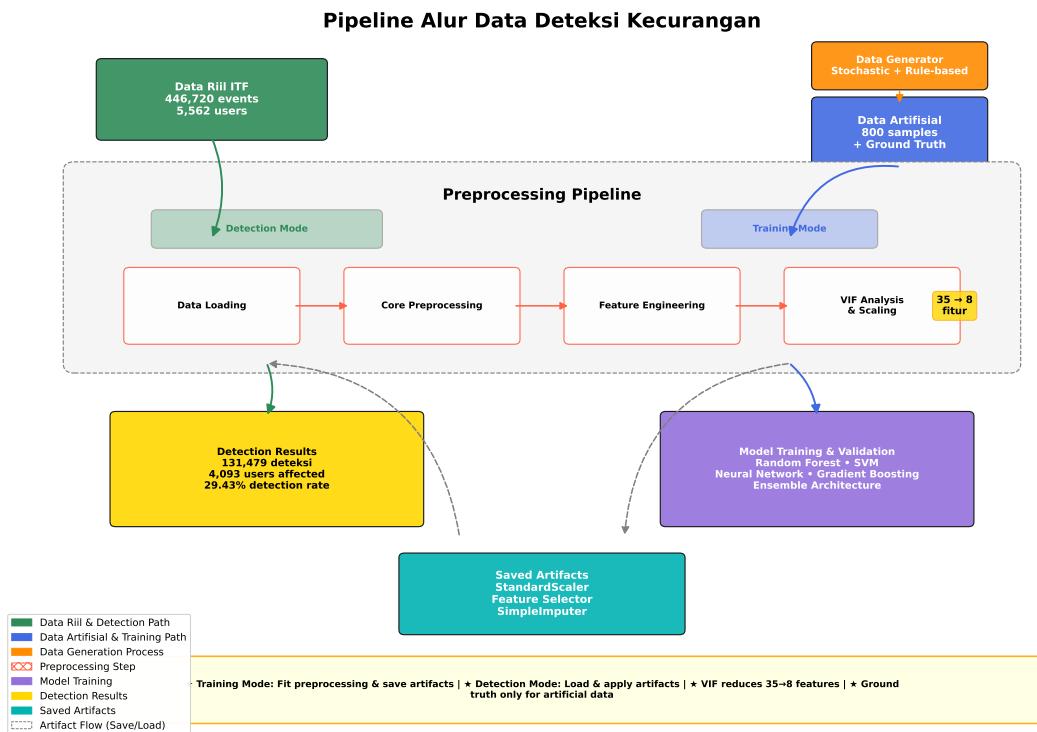
Bab ini memaparkan metode penelitian yang digunakan untuk mengembangkan sistem deteksi kecurangan akademik berbasis kecerdasan buatan. Pembahasan mencakup desain penelitian, tahapan metodologis yang ditempuh, sumber data yang digunakan, serta prosedur pengolahan dan analisis data. Metode penelitian dirancang untuk menjawab pertanyaan penelitian melalui pendekatan eksperimental yang sistematis dan terstruktur.



Gambar 3.1: Kerangka Metodologi Penelitian Deteksi Kecurangan

Gambar 3.1 menunjukkan kerangka metodologi penelitian yang terdiri dari enam fase utama yang saling berkaitan. Metodologi dirancang dengan pendekatan sistematis yang memungkinkan validasi dan reproducibility hasil penelitian. Setiap fase dirancang untuk menjawab aspek-aspek tertentu dari research questions yang telah dirumuskan pada Bab 1.

Gambar 3.2 menerjemahkan kerangka metodologi tingkat tinggi menjadi implementasi teknis yang spesifik. Diagram ini menampilkan alur teknis pemrosesan data dengan transparansi penuh mengenai sumber data dan perbedaan pemrosesan antara data riil dan artifisial.



Gambar 3.2: Pipeline Alur Data Deteksi Kecurangan

Pipeline dimulai dari dua sumber data: **Data Riil ITF** berisi 446,720 events dari aktivitas nyata mahasiswa, dan **Data Artifisial** dengan 800 sampel terkontrol beserta ground truth. Kedua jenis data diproses melalui **Preprocessing Pipeline** yang terdiri dari empat modul terintegrasi: Data Loading, Core Preprocessing, Feature Engineering, dan VIF & Scaling.

Perbedaan fundamental terletak pada mode operasi dual-processing: data artifisial menggunakan *Training Mode* untuk membangun dan menyimpan artifacts (scaler, feature selector, imputer), sedangkan data riil menggunakan *Detection Mode* dengan memanfaatkan artifacts yang telah disimpan. Strategi ini memastikan konsistensi transformasi dan mencegah data leakage. Proses feature engineering berhasil mereduksi 35 fitur awal menjadi 8 fitur stabil melalui analisis VIF. Pipeline menghasilkan model ensemble terlatih dan deteksi 131,479 percobaan ujian yang terindikasi kecurangan dengan tingkat deteksi 29.43

Kombinasi kedua diagram ini memberikan pemahaman komprehensif: Gambar 3.1 menyediakan roadmap metodologis untuk memandu pembaca melalui setiap bagian Bab 3, sedangkan Gambar 3.2 memberikan detail implementasi teknis yang memungkinkan reproduksi penelitian.

3.1 Kerangka Metodologi Penelitian

Penelitian ini mengadopsi pendekatan eksperimental dengan kombinasi data artifisial dan riil untuk mengembangkan sistem deteksi kecurangan yang robust. Desain penelitian dirancang untuk menjawab pertanyaan penelitian utama: bagaimana mengembangkan sistem deteksi kecurangan otomatis yang akurat untuk platform Moodle menggunakan pendekatan machine learning?

3.1.1 Tahapan Penelitian

Penelitian dilakukan melalui beberapa tahapan sistematis:

1. **Analisis Kebutuhan dan Studi Literatur:** Identifikasi pola-pola kecurangan akademik dalam pembelajaran daring dan review metode deteksi yang telah ada.
2. **Perancangan Data Artifisial:** Pengembangan generator data artifisial yang dapat mensimulasikan berbagai skenario kecurangan dengan parameter yang terkontrol.
3. **Pengembangan Pipeline:** Implementasi sistem end-to-end dari pra-pemrosesan data hingga deteksi kecurangan.
4. **Eksperimen dan Optimasi:** Pelatihan model dengan berbagai konfigurasi dan optimasi hyperparameter untuk mencapai performa optimal.
5. **Evaluasi dan Validasi:** Pengujian model pada data artifisial dengan ground truth dan aplikasi pada data riil untuk validasi praktis.

Setiap fase dalam kerangka metodologi yang ditampilkan pada Gambar ?? dirancang dengan pendekatan sistematis yang memungkinkan validasi dan reproducibility hasil penelitian. Keunggulan dari desain ini adalah transparansi penuh mengenai alur data yang berbeda antara mode training dan detection, sehingga memberikan pemahaman yang jelas tentang bagaimana sistem beroperasi dalam kondisi riil. Setiap fase dalam kerangka ini akan diuraikan secara detail dalam bagian-bagian berikutnya untuk memberikan pemahaman komprehensif tentang proses penelitian yang dilakukan.

3.1.2 Hipotesis Penelitian

Penelitian ini didasarkan pada hipotesis bahwa pola kecurangan akademik dalam ujian daring dapat diidentifikasi melalui analisis multi-dimensi dari data log aktivitas pengguna. Secara spesifik:

- Kolaborasi kecurangan akan menghasilkan pola similaritas yang dapat dideteksi dalam navigasi, timing, dan jawaban dengan tingkat akurasi tinggi ($\geq 90\%$).
- Ensemble model akan memberikan performa superior dibandingkan model tunggal dalam mendeteksi berbagai strategi kecurangan.
- Dataset dengan minimal 500-1000 sampel diperlukan untuk mencapai performa deteksi yang optimal.
- Model yang dilatih pada data artifisial dapat menggeneralisasi dengan baik pada data riil skala besar.

3.2 Studi Literatur dan Analisis Gap

Studi literatur terkait materi-materi yang relevan dengan penelitian dilakukan untuk menunjang penelitian. Topik-topik relevan yang dipelajari ialah *Moodle*, *Machine Learning*, Deteksi non-compliance, *Log Processing*, dan *eLearning*. Hasil dari studi literatur ini sebagai landasan dalam pengembangan model.

3.3 Akuisisi Data dan Perancangan Data Artifisial

Penelitian ini menggunakan pendekatan dual-data strategy yang menggabungkan data riil dan data artifisial untuk mengoptimalkan pengembangan model deteksi kecurangan. Strategi ini memungkinkan kontrol penuh terhadap ground truth dalam fase pelatihan sambil mempertahankan validitas real-world dalam fase evaluasi.

3.4 Pipeline Preprocessing Data

Sesuai dengan yang ditampilkan pada Phase 3 dalam Gambar ??, pipeline preprocessing data merupakan komponen krusial yang memproses kedua jenis data (riil dan artifisial) melalui empat modul terintegrasi. Pipeline ini dirancang untuk mendukung dua mode operasi yang berbeda: *training mode* untuk data artifisial dan *detection mode* untuk data riil.

3.4.1 Arsitektur Pipeline Preprocessing

Pipeline preprocessing terdiri dari empat modul yang saling terintegrasi:

Module I - Data Loading:

Modul ini bertanggung jawab untuk memuat dan memvalidasi data dari

berbagai tabel Moodle (mdl_quiz_attempts, mdl_question_attempt_steps, mdl_question_attempt_step_data, dll.) dengan skema yang telah didefinisikan. Validasi meliputi pemeriksaan tipe data, konsistensi referensial, dan penanganan data yang hilang atau corrupt.

Module II - Core Preprocessing:

Melakukan pembersihan dan normalisasi data fundamental, termasuk unifikasi timestamp ke format POSIX, penggabungan tabel-tabel terkait untuk membentuk event log yang komprehensif, dan filtering data sesuai kriteria penelitian (misalnya, hanya quiz attempts yang completed).

Module III - Feature Engineering:

Mengekstraksi fitur-fitur behavioral dari event log, mencakup:

- **Intra-Attempt Features:** Total duration, number of actions, step duration statistics
- **Sequential Features:** Navigation patterns, revisit counts, linearity measures, entropy
- **Similarity Features:** Inter-user similarity berdasarkan navigation, timing, dan answer patterns menggunakan Levenshtein distance dan korelasi statistik
- **Comparative Features:** Z-score normalization untuk membandingkan perilaku individual dengan rata-rata quiz

Module IV - Feature Post-Processing:

Menerapkan feature selection dan normalization lanjutan:

- **Missing Value Imputation:** Menggunakan strategi mean imputation dengan SimpleImputer
- **Collinearity Analysis:** Analisis VIF (Variance Inflation Factor) untuk mengidentifikasi dan menghilangkan fitur dengan multicollinearity tinggi ($VIF \geq 10$)
- **Variance Threshold Selection:** Menghilangkan fitur dengan variance rendah (threshold = 0.01)
- **Feature Scaling:** StandardScaler untuk normalisasi distribusi fitur

3.4.2 Dual Mode Processing Strategy

Perbedaan kunci dalam pemrosesan data riil dan artifisial terletak pada mode operasi pipeline:

Training Mode (Data Artifisial):

Data artifisial diproses dalam training mode dimana semua modul pipeline melakukan fitting terhadap data dan menyimpan artifacts (imputer, scaler, feature selector, selected feature lists) ke dalam direktori preprocessing/artifacts/. Mode ini memungkinkan eksplorasi dan optimasi parameter preprocessing untuk mencapai performa model optimal.

Detection Mode (Data Riil):

Data riil diproses dalam detection mode dimana pipeline menggunakan artifacts yang telah disimpan dari training mode. Hal ini memastikan konsistensi transformasi dan mencegah data leakage. Model preprocessing yang sama yang digunakan untuk training akan diterapkan secara identik pada data riil untuk detection.

Strategi dual mode ini memastikan bahwa model yang dilatih pada data artifisial dapat diterapkan secara valid pada data riil tanpa bias metodologis, sambil mempertahankan transparansi penuh dalam proses transformasi data.

Data Riil Moodle:

Data log *Moodle* diperoleh langsung dari sistem yang dikelola oleh tim ITF Fasilkom UI dan telah melalui proses anonimisasi untuk menjaga privasi pengguna. Data ini digunakan untuk validasi model dalam konteks operasional nyata.

Data Artifisial:

Data artifisial dirancang khusus untuk pelatihan model dengan ground truth yang terdokumentasi. Pendekatan ini memungkinkan eksplorasi berbagai skenario kecurangan dan kontrol parameter yang tidak mungkin dilakukan pada data riil.

3.4.3 Data Log Moodle Riil: Deskripsi (periode, jumlah event/user, fitur utama), Proses Akuisisi, Kebijakan Anonimisasi & Etika.

Subbab ini menjelaskan mengenai data log yang diambil langsung dari sistem *Moodle*, yang dikelola oleh tim ITF Fasilkom UI dan disimpan pada Lumbung Storage Cloud (mirip dengan platform penyimpanan seperti Google Drive). Data yang digunakan telah melalui proses anonimisasi, di mana identitas asli pengguna (username atau nama lengkap) tidak disertakan, melainkan hanya diwakili oleh *user_id*.

3.4.3.1 Deskripsi Data

Data log *Moodle* terdiri dari beberapa tabel utama yang masing-masing menyimpan informasi berbeda terkait aktivitas pengguna dan kuis. Berikut adalah rincian kolom-kolom yang terdapat dalam tiap tabel:

mdl_question_usages

Kolom: *question_usage_id, context_id*

Menyimpan informasi terkait konteks penggunaan pertanyaan dalam kuis.

mdl_quiz_grades

Kolom: *quiz_grades_id, quiz_id, user_id, final_grade*

Berisi nilai akhir dari masing-masing kuis yang diambil oleh pengguna.

mdl_question_attempt_steps

Kolom: *question_step_id, question_attempt_id, sequencenumber, state, timecreated*

Mencatat tiap langkah (*step*) dalam upaya penggerjaan soal, termasuk status dan waktu pembuatan.

mdl_quiz_attempts

Kolom: *attempt_id, quiz_id, user_id, question_usage_id, timestart, timefinish, state, sum-grades*

Menggambarkan detail setiap upaya penggerjaan kuis, termasuk waktu mulai, waktu selesai, status, dan jumlah nilai yang diperoleh.

Catatan: Terdapat anomali pada *timefinish* (misalnya timestamp 1970-01-01) yang kemungkinan menunjukkan upaya kuis yang belum selesai atau default nilai dari Unix epoch.

mdl_question_answers

Kolom: *question_answers_id, questionid, answer_text, fraction*

Menyimpan data mengenai jawaban yang diberikan pada tiap pertanyaan, termasuk teks jawaban dan bobot nilai yang terkait.

mdl_question_attempt_step_data

Kolom: *step_data_id, question_step_id, name, value*

Menyimpan data tambahan terkait langkah penggerjaan soal, yang dapat berupa nilai-nilai

pendukung dari proses evaluasi.

mdl_quiz

Kolom: *quiz_id, course, quiz_name, timeopen, timeclose, timelimit*

Berisi informasi dasar mengenai kuis, termasuk mata kuliah, nama kuis, serta waktu buka dan tutup kuis.

mdl_sessions

Kolom: *session_id, user_id, timecreated, lastip, sessdata*

Mencatat aktivitas sesi pengguna, mulai dari waktu pembuatan sesi hingga informasi terkait IP dan data sesi lainnya.

3.4.3.2 Rentang Waktu dan Skala Dataset

Data log *Moodle* mencakup periode hampir 10 tahun, dengan rentang data keseluruhan dari tanggal 31 Juli 2015 hingga 22 Februari 2025. Secara spesifik:

mdl_question_attempt_steps:

Rentang waktu: 29 Agustus 2015 – 22 Februari 2025

Durasi: sekitar 3.565 hari

mdl_quiz_attempts:

Rentang waktu untuk *timestart*: 31 Juli 2015 – 22 Februari 2025

Rentang waktu untuk *timefinish*: (dengan catatan anomali timestamp 1970 sebagai default) hingga 22 Februari 2025

Durasi: sekitar 3.594 hari (mengabaikan nilai default 1970)

mdl_sessions:

Rentang waktu: 17 Maret 2020 – 22 Februari 2025

Durasi: sekitar 1.803 hari

Skala data secara keseluruhan meliputi:

- Total kuis yang diambil: 446.720 upaya
- Jumlah pengguna unik: 5.562
- Jumlah kuis unik: 6.304
- Jumlah langkah pertanyaan: 22.192.809

3.4.3.3 Cakupan Mata Kuliah dan Pola Penggunaan

Data ini mencakup aktivitas di lebih dari 140 Mata Kuliah unik, dengan variasi ukuran kelas:

Ukuran Mata Kuliah:

- Mata Kuliah besar (300+ mahasiswa): sekitar 10 Mata Kuliah
- Mata Kuliah menengah (100-300 mahasiswa): sekitar 30 Mata Kuliah
- Mata Kuliah kecil (<100 mahasiswa): sekitar 100 Mata Kuliah
- Mata Kuliah sangat kecil (<10 mahasiswa): sekitar 15 Mata Kuliah

Contoh Mata Kuliah dengan aktivitas tinggi:

- Mata Kuliah 3836: 453 peserta, 5.544 upaya (rata-rata 12,24 upaya per pengguna)
- Mata Kuliah 3634: 442 peserta, 5.902 upaya (rata-rata 13,35 upaya per pengguna)
- Mata Kuliah 3723: 390 peserta, 3.829 upaya (rata-rata 9,56 upaya per pengguna)
- Mata Kuliah 3640: 386 peserta, 4.463 upaya (rata-rata 11,56 upaya per pengguna)
- Mata Kuliah 3636: 382 peserta, 761 upaya (rata-rata 1,99 upaya per pengguna)

Pola penggunaan kuis menunjukkan perbedaan yang signifikan antar Mata Kuliah. Beberapa Mata Kuliah hanya mencatat satu upaya per pengguna, sedangkan Mata Kuliah lain mencatat rata-rata 10 upaya atau lebih, yang mengindikasikan adanya kuis latihan atau kebijakan pengulangan untuk meningkatkan pemahaman materi.

3.4.3.4 Proses Akuisisi dan Kebijakan Anonimisasi

Data log *Moodle* diakuisisi secara langsung dari sistem *Moodle* oleh tim ITF Fasilkom UI dan kemudian disimpan di Lumbung Storage Cloud. Proses akuisisi melibatkan:

Pengambilan Data:

Data diekstraksi dari server *Moodle* dengan menggunakan prosedur logging yang telah ditetapkan, memastikan setiap aktivitas terekam secara lengkap.

Pengiriman dan Penyimpanan:

Data dikirim dan disimpan secara terpusat di Lumbung Storage Cloud yang merupakan repository internal Fasilkom UI, menjamin keamanan dan integritas data.

Anonimisasi:

Untuk melindungi privasi pengguna, data telah diproses sehingga informasi identitas pribadi (username, nama lengkap, dan data sensitif lainnya) dihilangkan. Hanya *user_id* yang tetap dipertahankan untuk keperluan analisis. Hal ini sesuai dengan standar etika penelitian dan kebijakan perlindungan data yang berlaku.

Pertimbangan Etika:

Penggunaan data riil ini telah mempertimbangkan aspek etika dan regulasi perlindungan data. Kebijakan anonimisasi yang diterapkan memastikan bahwa data tidak dapat dikaitkan langsung dengan individu tertentu, sehingga menjaga kerahasiaan dan privasi pengguna.

3.4.4 Strategi Data Artifisial

Data artifisial dalam penelitian ini dirancang untuk mengatasi keterbatasan ground truth pada data riil. Strategi ini memungkinkan kontrol penuh terhadap parameter kecurangan dan validasi objektif terhadap performa model, dengan alasan-alasan sebagai berikut:

1. Kontrol Variabel dan Simulasi Skenario Ekstrem:

Data artifisial memungkinkan penciptaan skenario perilaku pengguna yang ekstrem atau tidak biasa, yang mungkin jarang terjadi pada data log riil. Dengan demikian, model dapat dilatih untuk mengenali pola-pola *non-compliance* secara lebih spesifik dan mendalam.

2. Pengembangan Model yang Terarah:

Dalam penelitian ini, pengembangan model dilakukan sepenuhnya dengan menggunakan data artifisial. Pendekatan ini memberikan fleksibilitas dalam mengatur parameter dan iterasi pelatihan, serta mengurangi risiko *overfitting* pada data riil yang belum terlabel dengan jelas.

3. Evaluasi Model dengan Data Riil:

Setelah model dioptimasi dengan data artifisial, evaluasi akhir akan dilakukan dengan menerapkan model pada data log asli. Langkah ini bertujuan untuk mengukur sejauh mana model dapat mendeteksi kasus kecurangan yang benar-benar terjadi di lingkungan sistem *Moodle*, sehingga memberikan validasi nyata terhadap efektivitas pendekatan yang digunakan.

4. Pendekatan Metodologis dalam Pembuatan Data Artifisial:

Data artifisial dihasilkan melalui kombinasi simulasi berbasis aturan dan proses

stokastik, yang disesuaikan dengan pola penggunaan yang ditemukan dalam data log riil. Pendekatan ini memastikan bahwa data artifisial tidak hanya mereplikasi kondisi normal, tetapi juga memasukkan skenario *non-compliance* yang relevan untuk pengembangan model.

Dengan demikian, penggunaan data artifisial dalam penelitian ini memberikan keuntungan dalam hal kontrol variabilitas dan eksplorasi skenario ekstrem, serta mempercepat proses pengembangan model. Evaluasi akhir dengan data log riil akan menjadi tolok ukur untuk menilai kemampuan model dalam mendeteksi kecurangan yang terjadi secara nyata dalam sistem *Moodle*.

3.5 Preprocessing Pipeline dan Feature Engineering

Tahap feature engineering dan analisis VIF merupakan komponen kritis dalam pipeline penelitian ini. Proses ini mencakup transformasi data mentah menjadi representasi fitur yang bermakna, analisis multikolinearitas, dan seleksi fitur optimal untuk model machine learning.

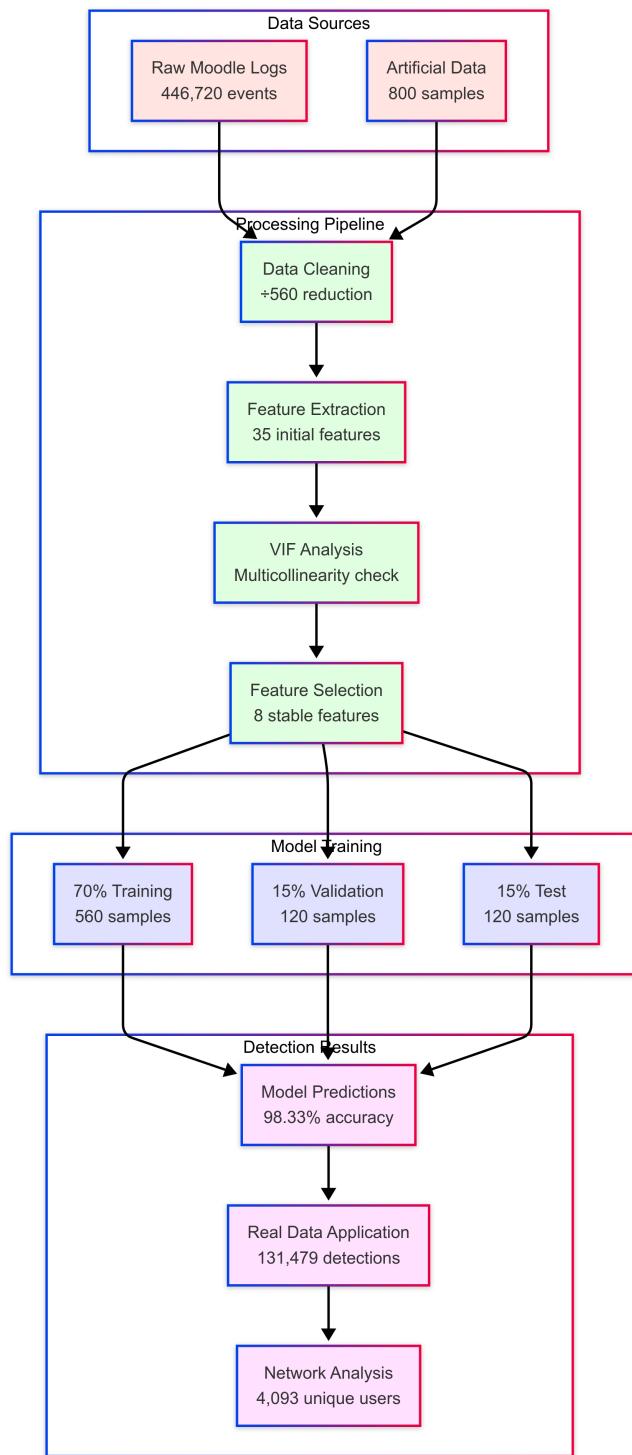
Gambar 3.3 menunjukkan alur transformasi data dari raw logs hingga detection output. Reduksi dramatis dari 446,720 events menjadi 800 training samples (faktor 560x) dilakukan melalui agregasi per user-quiz attempt. Feature extraction menghasilkan 35 fitur awal yang kemudian direduksi menjadi 8 fitur stabil melalui VIF analysis. Data dibagi dengan proporsi 70/15/15 untuk training, validation, dan test untuk memastikan evaluasi yang fair dan mencegah overfitting.

Proses pra-pemrosesan tidak hanya bertujuan untuk mengurangi noise dan inkonsistensi, tetapi juga untuk memastikan bahwa fitur-fitur yang dihasilkan mencerminkan karakteristik perilaku pengguna secara akurat. Langkah-langkah berikut diambil dengan dasar metodologis yang defendable secara ilmiah:

3.5.1 Pembersihan Data (Data Cleaning)

Penanganan Missing Values:

Data log sering kali mengandung nilai yang hilang (*missing values*) karena ketidakteraturan dalam pencatatan event atau error saat pengambilan data. Dalam pipeline, nilai yang tidak terisi diimputasi dengan metode pengantian menggunakan nilai default (misalnya,



Gambar 3.3: Alur Data dalam Pipeline Deteksi Kecurangan

0) atau perhitungan statistik (rata-rata/median) ketika relevan. Misalnya, dalam skrip `preprocess_features.py`, setelah ekstraksi fitur, seluruh nilai NaN diisi dengan nol untuk memastikan tidak ada celah data yang dapat mengganggu analisis model.

Filtering Data yang Tidak Relevan:

Dalam konteks deteksi kecurangan, tidak seluruh event log memiliki nilai informasi yang sama. Oleh karena itu, dilakukan penyaringan untuk:

- Mengabaikan event dengan atribut tertentu (misalnya, field *contextlevel* yang bernilai 'system') karena event ini tidak mewakili aktivitas pengguna pada kuis.
- Menghapus entri dengan *user_id* yang null, mengingat identifikasi pengguna merupakan variabel kunci dalam analisis pola perilaku.

3.5.2 Transformasi dan Normalisasi Data

Unifikasi Format Waktu dan Normalisasi Zona Waktu:

Data log mengandung timestamp yang berasal dari sumber atau zona waktu yang berbeda. Untuk memastikan keseragaman, seluruh timestamp dikonversi ke dalam format numerik, seperti Unix timestamp (atau ISO 8601 jika diperlukan), melalui fungsi konversi di `preprocess_features.py`. Proses ini juga melibatkan normalisasi zona waktu agar seluruh event log dapat dibandingkan secara akurat dalam kerangka waktu yang sama.

Parsing Nested Fields:

Banyak kolom dalam data log disimpan dalam bentuk string yang merepresentasikan array atau dictionary (misalnya, sequence navigasi atau transition times). Menggunakan fungsi seperti `ast.literal_eval`, skrip melakukan parsing terhadap string tersebut sehingga struktur data yang tersusun (list atau dictionary) dapat diekstraksi. Proses ini memungkinkan perhitungan fitur statistik seperti panjang sequence, entropi, dan jumlah revisits, yang esensial untuk mendeteksi pola aktivitas pengguna.

3.5.3 Ekstraksi Fitur dan Deteksi Outlier

Setelah data dibersihkan dan ditransformasikan, tahap selanjutnya adalah ekstraksi fitur, di mana fitur-fitur dasar dan lanjutan dihasilkan untuk mendukung proses pelatihan model deteksi kecurangan. Dalam pipeline, ekstraksi fitur dilakukan melalui modul `feature_eng.py` dan `fixed_extraction.py`, dengan beberapa langkah sebagai berikut:

Fitur Dasar:

Fitur seperti jumlah percobaan kuis, rata-rata waktu pengerjaan, total waktu, serta statistik minimum dan maksimum dihitung dari data log kuis. Contoh implementasi terdapat pada fungsi `extract_basic_statistics()` di `feature_eng.py`, yang juga menyertakan agregasi data per pasangan user-kuis.

Fitur Sequence:

Dari data navigasi dan jawaban, diekstraksi fitur-fitur seperti:

- Panjang sequence dan jumlah pertanyaan unik: Mengukur seberapa panjang dan bervariasinya aktivitas navigasi pengguna.
- Linearity: Menghitung seberapa berurutan langkah-langkah yang diambil, menggunakan perhitungan rasio antara pertanyaan unik dan total langkah.
- Revisits: Penghitungan jumlah revisits atau langkah yang diulang, sebagai indikasi pola abnormal.

Fungsi `extract_navigation_features()` dalam `fixed_extraction.py` mengilustrasikan bagaimana fitur-fitur tersebut dihitung dengan memanfaatkan evaluasi list dari sequence.

Deteksi Outlier pada Fitur Waktu:

Pada ekstraksi fitur waktu, dilakukan perhitungan statistik seperti rata-rata, standar deviasi, nilai minimum dan maksimum dari durasi pengerjaan soal. Selain itu, skrip juga menghitung jumlah event dengan durasi sangat pendek (misalnya, ≤ 5 detik) dan sangat panjang (misalnya, ≥ 600 detik).

Berikut adalah cuplikan kode dari fungsi `extract_timing_features()` yang digunakan untuk mendeteksi pola waktu yang abnormal, yang dapat dijadikan indikator outlier:

```
def extract_timing_features(time_seq):
    """Extract features from timing sequence."""
    features = {}

    try:
        if isinstance(time_seq, str):
            time_seq = [float(t) for t in time_seq.split()]
        else:
            time_seq = [float(t) for t in time_seq]
    except ValueError:
        return None

    # calculate basic statistics
    features['mean'] = np.mean(time_seq)
    features['std'] = np.std(time_seq)
    features['min'] = np.min(time_seq)
    features['max'] = np.max(time_seq)
    features['q1'] = np.percentile(time_seq, 25)
    features['q3'] = np.percentile(time_seq, 75)
    features['iqr'] = features['q3'] - features['q1']

    # calculate outliers
    lower_bound = features['q1'] - 1.5 * features['iqr']
    upper_bound = features['q3'] + 1.5 * features['iqr']
    outliers = [t for t in time_seq if t < lower_bound or t > upper_bound]

    features['outliers'] = len(outliers)

    return features
```

```

        time_seq = eval(time_seq)
        time_seq = np.array(time_seq, dtype=float)

        features['mean_time'] = float(np.mean(time_seq))
        features['std_time'] = float(np.std(time_seq))
        features['min_time'] = float(np.min(time_seq))
        features['max_time'] = float(np.max(time_seq))

        # Suspicious timing patterns sebagai indikator outlier
        features['quick_answers'] = int(sum(1 for t in time_seq if t < 5))
        features['very_long_answers'] = int(sum(1 for t in time_seq if t > 600))
    except Exception as e:
        print(f"Warning: Error processing timing features: {e}")
        features['mean_time'] = 0.0
        features['std_time'] = 0.0
        features['min_time'] = 0.0
        features['max_time'] = 0.0
        features['quick_answers'] = 0
        features['very_long_answers'] = 0

    return features

```

Nilai `quick_answers` dan `very_long_answers` ini memberikan gambaran tentang berapa kali pengguna menyelesaikan bagian tertentu dengan durasi yang sangat tidak wajar, yang kemudian dapat diintegrasikan sebagai fitur untuk mendeteksi perilaku kecurangan.

Penghitungan Similarity Features:

Untuk mendeteksi kolaborasi kecurangan, pipeline menghitung matriks kemiripan (*similarity matrix*) berdasarkan pola navigasi, waktu, dan jawaban antar pengguna. Fungsi `calculate_similarity_matrices()` di `feature_eng.py` menghitung kemiripan menggunakan metode seperti Levenshtein distance untuk sequence navigasi dan korelasi untuk sequence waktu, kemudian hasilnya digunakan untuk menambahkan fitur agregat seperti rata-rata dan maksimum similarity antar pengguna.

3.5.4 Checklist Pra-pemrosesan

Untuk memastikan bahwa seluruh proses pra-pemrosesan dapat diulangi dan diverifikasi, disusunlah checklist sebagai berikut:

- Konversi seluruh timestamp ke format standar (Unix timestamp atau ISO 8601).
- Penghapusan event dengan atribut *contextlevel* bernilai 'system'.
- Penyaringan entri dengan *user_id* null.
- Parsing kolom yang berisi string representasi array/dict untuk ekstraksi fitur.
- Normalisasi zona waktu untuk keseragaman data.
- Perhitungan statistik fitur dasar dan sequence (termasuk deteksi pola outlier pada waktu).

3.5.5 Justifikasi Ilmiah

Pendekatan pra-pemrosesan yang diterapkan dalam penelitian ini didasarkan pada prinsip-prinsip *data cleaning* dan *feature engineering* yang telah terbukti secara empiris meningkatkan kualitas dataset dan kinerja model *machine learning*.

Pembersihan dan Transformasi:

Dengan memastikan bahwa data dalam format yang konsisten dan bebas dari *missing values*, variabilitas yang tidak relevan dapat diminimalkan, sehingga model tidak terdistorsi oleh noise data.

Ekstraksi Fitur:

Fitur-fitur yang diekstraksi, seperti linearity dan revisits pada sequence, memberikan representasi numerik yang dapat menggambarkan perilaku pengguna secara mendalam.

Deteksi Outlier:

Dengan mengidentifikasi event dengan durasi yang sangat singkat atau sangat panjang, pipeline mampu memberikan sinyal peringatan terhadap kemungkinan aktivitas yang tidak wajar, yang secara langsung berkontribusi pada identifikasi kecurangan.

Reproducibility:

Checklist dan struktur modular pada skrip memastikan bahwa seluruh proses dapat direplikasi dan diaudit, sehingga mendukung validitas dan reproducibility dari penelitian.

Melalui serangkaian proses yang sistematis dan berbasis algoritma yang telah teruji, tahap

pra-pemrosesan ini memberikan dasar yang kuat untuk langkah-langkah selanjutnya dalam pipeline, yaitu ekstraksi fitur lanjutan, pelatihan model, dan akhirnya evaluasi deteksi kecurangan.

3.5.6 Perancangan dan Generasi Data Artifisial

Subbab ini menjelaskan secara mendalam mengenai rancangan, implementasi, dan validasi data log *Moodle* artifisial yang digunakan untuk mengembangkan model deteksi *non-compliance*. Pendekatan yang diterapkan dikenal dengan istilah *Skenario Perilaku Sintetik*, yaitu simulasi aktivitas pengguna (baik perilaku normal maupun *non-compliance*) melalui algoritma yang menggabungkan simulasi berbasis aturan dan proses stokastik. Data artifisial ini tidak hanya mereplikasi aktivitas log riil, tetapi juga memungkinkan eksplorasi skenario ekstrem yang jarang terekam pada data nyata, seperti koordinasi kelompok kecurangan dengan pola sinkronisasi tinggi.

3.5.7 Definisi Operasional Skenario Perilaku Sintetik

Skenario Perilaku Sintetik didefinisikan sebagai rangkaian aturan dan mekanisme yang dirancang untuk mereplikasi pola aktivitas pengguna pada sistem *Moodle*, baik dalam kondisi normal maupun *non-compliance* (kecurangan). Pendekatan ini mengintegrasikan simulasi berbasis aturan dengan proses stokastik, sehingga memungkinkan pembentukan data log artifisial yang tidak hanya mereplikasi aktivitas log riil, tetapi juga dapat mengeksplorasi skenario ekstrem yang jarang terekam dalam data nyata.

Perilaku Normal:

Pada skenario perilaku normal, aktivitas pengguna dirancang untuk mencerminkan dinamika berpikir yang alami, yang ditandai dengan:

- Variasi Urutan Navigasi: Pengguna normal menunjukkan urutan akses pertanyaan yang bervariasi, dengan kemungkinan revisi jawaban yang berbeda-beda antar sesi. Hal ini menggambarkan proses evaluasi internal yang dinamis.
- Variasi Pola Jawaban: Pola jawaban mencerminkan respon acak yang wajar, dengan proporsi jawaban benar dan salah yang bervariasi secara natural.
- Waktu Pengerjaan yang Variatif: Interval waktu antar pertanyaan bervariasi, mencerminkan kecepatan berpikir dan penyesuaian terhadap tingkat kesulitan pertanyaan. Distribusi waktu pengerjaan ini umumnya menunjukkan standar deviasi yang wajar.

Perilaku Non-Compliance (Kecurangan):

Pada skenario *non-compliance*, pola aktivitas sengaja diatur untuk menciptakan indikasi kecurangan, dengan karakteristik sebagai berikut:

- Kesamaan Navigasi: Urutan pertanyaan yang diakses oleh pengguna dalam kelompok kecurangan sangat mirip, termasuk adanya revisi yang identik antar anggota kelompok.
- Konsistensi Jawaban: Pola jawaban menunjukkan tingkat keseragaman yang tinggi. Misalnya, terdapat kecenderungan anggota kelompok secara bersama-sama memberikan jawaban salah pada pertanyaan-pertanyaan sulit.
- Sinkronisasi Waktu yang Tidak Wajar: Interval waktu antar pertanyaan hampir seragam antar pengguna. Dalam beberapa kasus, terdapat kelompok yang menyelesaikan kuis secara bersamaan dalam waktu kurang dari 15 detik, dengan standar deviasi yang rendah, mengindikasikan adanya pola sinkronisasi yang sulit terjadi secara natural.
- Mekanisme Leader-Follower: Terdapat pola di mana satu anggota (leader) menyelesaikan kuis terlebih dahulu, sedangkan anggota lainnya (follower) mengikuti dengan delay yang konsisten. Pola ini menciptakan korelasi tinggi dalam waktu pengerjaan antar anggota, yang menjadi indikator kuat adanya koordinasi.

Definisi operasional ini menjadi dasar untuk memformulasikan parameter simulasi. Parameter-parameter tersebut, seperti tingkat kemiripan navigasi, delay waktu, dan distribusi jawaban, kemudian diintegrasikan dalam algoritma generasi data log artifisial. Setiap skenario diberikan label *ground truth*, yang memungkinkan validasi dan evaluasi model deteksi kecurangan secara kuantitatif dan kualitatif.

3.5.8 Desain Ground Truth Artifisial

Dalam proses generasi data log artifisial, setiap entitas baik pada level sesi, upaya pengerjaan, maupun langkah pertanyaan diberi label *ground truth* yang mendefinisikan status sebagai aktivitas normal atau *non-compliance*. Desain *ground truth* ini tidak hanya merupakan hasil dari parameter simulasi, tetapi juga dilengkapi dengan dokumentasi yang komprehensif melalui file `cheating_ground_truth.md`. File ini berfungsi sebagai acuan empiris untuk validasi dan evaluasi model deteksi kecurangan.

Komponen Utama Ground Truth:

- **Label Kecurangan:**

Setiap entitas data dilengkapi dengan label:

- 0: Menunjukkan aktivitas normal.
- 1: Menunjukkan aktivitas *non-compliance* (kecurangan), yang dihasilkan dari simulasi kelompok dengan pola sinkronisasi tinggi dan koordinasi yang jelas.

- **Komposisi Dataset:**

Data artifisial dihasilkan dengan proporsi tertentu antara perilaku normal dan *non-compliance*. Proporsi ini dikonfigurasi dalam parameter simulasi, misalnya 1020% pengguna disimulasikan sebagai *non-compliance*, untuk memastikan keseimbangan yang memadai dalam pelatihan dan evaluasi model.

- **Pencatatan Parameter Simulasi dan Contoh Kasus:**

Parameter-parameter yang mendasari penetapan label *ground truth* direkam secara detail, mencakup:

- Navigation Similarity: Persentase kesamaan urutan navigasi antar anggota kelompok (contoh: 96%).
- Answer Pattern Similarity: Persentase kesamaan pola jawaban (contoh: 94%).
- Timing Correlation: Koefisien korelasi waktu antar pertanyaan (contoh: 0.95).
- Standard Deviation (Avg): Rata-rata standar deviasi waktu pengerjaan per pertanyaan (contoh: 12 detik).
- Wrong Answer Bias: Probabilitas kesalahan terkoordinasi (contoh: 87%).

Nilai-nilai di atas disajikan sebagai contoh dalam file `cheating_ground_truth.md`. Penting untuk dicatat bahwa angka-angka tersebut merupakan representasi sample yang dapat dikonfigurasi ulang sesuai kebutuhan eksperimen, dan bukan merupakan nilai final yang harus diterapkan secara universal.

Dokumentasi Melalui File Cheating Ground Truth:

File `cheating_ground_truth.md` menyajikan tabel ringkasan statistik kelompok kecurangan beserta parameter-parameter yang telah diukur secara simulatif. Contoh tabel tersebut mencakup:

- Kelompok Kecurangan dengan Severity Tinggi: Menunjukkan kesamaan navigasi, pola jawaban, dan korelasi waktu yang sangat tinggi, serta standar deviasi dan wrong answer bias yang rendah.
- Kelompok Kecurangan dengan Severity Sedang: Menunjukkan nilai yang lebih moderat,

dengan delay waktu dan variansi yang lebih besar.

Dokumentasi ini menyediakan bukti empiris dan justifikasi statistik bahwa skenario *non-compliance* yang disimulasikan memiliki karakteristik yang berbeda secara signifikan dari aktivitas normal. Informasi ini sangat penting untuk validasi model, karena memungkinkan perbandingan langsung antara prediksi model dengan *ground truth*.

Integrasi dan Reproducibility:

Ground truth artifisial yang terdokumentasi secara rinci ini diintegrasikan langsung ke dalam dataset log artifisial. Pendekatan ini memastikan bahwa evaluasi model dapat dilakukan secara objektif menggunakan metrik seperti precision, recall, F1-score, dan akurasi. Selain itu, pencatatan parameter simulasi dan contoh kasus dalam file *cheating_ground_truth.md* mendukung *reproducibility* penelitian, karena eksperimen dapat diulang dengan menggunakan seed dan konfigurasi yang sama.

Dengan demikian, desain *ground truth* artifisial ini tidak hanya menyediakan label yang diperlukan untuk evaluasi model, tetapi juga memberikan kerangka kerja untuk analisis sensitivitas dan validasi statistik, yang secara bersama-sama mendukung pendekatan *Skenario Perilaku Sintetik* yang digunakan dalam penelitian ini.

3.5.9 Metode Generasi Data

Proses generasi data log artifisial dilakukan dengan pendekatan yang menggabungkan simulasi berbasis aturan dan proses stokastik, sehingga dapat mereplikasi pola aktivitas pengguna pada sistem *Moodle* secara realistik. Pendekatan ini juga memungkinkan eksplorasi skenario ekstrem *non-compliance* yang jarang terekam pada data riil. Metode yang diterapkan meliputi beberapa komponen kunci berikut:

Simulasi Berbasis Aturan:

Algoritma simulasi menetapkan seperangkat aturan untuk menentukan urutan navigasi, pola revisi jawaban, dan distribusi waktu antar pertanyaan. Aturan-aturan ini dirancang untuk meniru dinamika aktivitas pengguna normal maupun perilaku kecurangan. Sebagai contoh, pada kelompok *non-compliance*, aturan mensyaratkan agar setiap anggota mengikuti urutan pertanyaan yang seragam dan menunjukkan kesamaan dalam revisi jawaban, sehingga menghasilkan tingkat kesamaan navigasi dan pola jawaban yang tinggi.

Proses Stokastik:

Untuk menghindari keteraturan yang sempurna dan menambahkan tingkat realisme, proses stokastik diterapkan dalam penentuan waktu penggerjaan, revisi jawaban, dan urutan pertanyaan. Penggunaan fungsi `random` dengan seed yang telah ditentukan memungkinkan variasi terkontrol, sehingga setiap simulasi menghasilkan distribusi waktu dan pola yang mendekati kondisi riil, namun masih mempertahankan karakteristik khas dari skenario *non-compliance*.

Modifikasi Berdasarkan Data Riil:

Parameter dasar seperti `base_date`, `timelimit`, serta distribusi waktu penggerjaan diadaptasi dari analisis data log *Moodle* riil. Pendekatan ini memastikan bahwa data artifisial tidak hanya bersifat syntetik, tetapi juga memiliki kemiripan statistik dengan data riil. Dengan demikian, model yang dikembangkan dapat diuji dan divalidasi secara lebih representatif terhadap kondisi operasional di lingkungan nyata.

Penerapan Mekanisme Leader-Follower:

Untuk mengantisipasi skenario kecurangan dengan tingkat koordinasi yang tinggi, algoritma juga mengimplementasikan pola leader-follower. Dalam mekanisme ini, satu anggota (*leader*) menyelesaikan kuis terlebih dahulu dengan pola waktu yang stabil, sementara anggota lain (*follower*) mengikuti dengan delay konsisten yang telah disimulasi. Penghitungan korelasi waktu antar anggota yang tinggi (misalnya, nilai $\zeta > 0.8$) menjadi indikator kuat atas adanya koordinasi, sehingga pola ini diintegrasikan dalam parameter simulasi.

Parameter Simulasi dan Konfigurasi:

Seluruh proses generasi data dikonfigurasi melalui file konfigurasi (misalnya, file JSON) yang menetapkan parameter-parameter penting, seperti:

- Jumlah total pengguna dan kuis.
- Proporsi pengguna dengan perilaku *non-compliance*.
- Batasan waktu (`timelimit`) dan `base_date` sebagai dasar simulasi.
- Nilai threshold untuk *similarity* dalam pola navigasi, jawaban, dan waktu.

Parameter-parameter ini bersifat fleksibel dan dapat disesuaikan untuk mengeksplorasi berbagai skenario, mulai dari aktivitas normal hingga skenario ekstrem yang memicu

koordinasi *non-compliance* secara signifikan.

Melalui kombinasi simulasi berbasis aturan dan proses stokastik, metode generasi data ini mampu menghasilkan log aktivitas yang kaya informasi dan mendekati realitas, sekaligus memungkinkan pengujian model dalam berbagai kondisi ekstrem. Pendekatan ini memberikan dasar yang solid untuk evaluasi dan validasi model deteksi *non-compliance*, dengan memastikan bahwa data artifisial yang dihasilkan mencerminkan variasi dan dinamika yang terjadi pada data log riil.

3.5.10 Implementasi Teknis

Implementasi teknis dalam generasi data log artifisial dilakukan dengan menggunakan bahasa pemrograman Python, yang mendukung replikasi dan validitas eksperimen melalui pendekatan yang modular dan terdokumentasi dengan baik. Proses ini memastikan bahwa data artifisial yang dihasilkan memiliki struktur dan karakteristik yang konsisten dengan data log *Moodle* riil. Komponen-komponen teknis utama yang digunakan adalah sebagai berikut:

Platform dan Library Utama:

- Python: Dipilih karena fleksibilitas dan ekosistemnya yang luas dalam pengolahan data dan simulasi.
- Pandas dan Numpy: Digunakan untuk manipulasi data dan komputasi numerik, dengan data log disimpan serta diproses dalam format CSV untuk memudahkan integrasi ke pipeline selanjutnya.
- Faker: Digunakan untuk menghasilkan data pengguna yang realistik, seperti *username*, nama, dan informasi identitas lain yang mendekati kondisi riil.
- Random dan Modul Stokastik: Fungsi `random`, yang dijalankan dengan seed tertentu, memastikan variasi terkontrol dalam waktu pengerjaan, urutan navigasi, dan pola jawaban. Hal ini memungkinkan simulasi yang konsisten dan dapat direproduksi.
- JSON dan CSV: File konfigurasi, *ground truth*, dan output data disimpan dalam format JSON dan CSV sehingga mendukung dokumentasi dan analisis statistik.

Modularitas Kode:

Implementasi dibangun secara modular untuk mendukung pemeliharaan dan pengem-

bangun lebih lanjut. Modul utama yang terintegrasi dalam pipeline meliputi:

- `generate_case.py`: Modul inti yang menghasilkan data log artifisial berdasarkan parameter yang telah ditetapkan. Modul ini mengintegrasikan simulasi berbasis aturan dan proses stokastik untuk menciptakan *Skenario Perilaku Sintetik*.
- `fixed_extraction.py` dan `preprocess_features.py`: Walaupun proses ekstraksi dan pembersihan fitur secara detail akan dibahas pada subbab 3.6, modul ini digunakan untuk memastikan bahwa data artifisial dapat diproses dengan standar yang sama seperti data log riil, sehingga mendukung integrasi ke tahap evaluasi model.

Reproducibility dan Konfigurasi:

Penggunaan Seed Random: Seluruh fungsi random dijalankan dengan seed yang telah ditentukan, sehingga eksperimen dapat diulang dengan hasil yang konsisten.

File Konfigurasi: Parameter simulasi seperti jumlah pengguna, proporsi *non-compliance*, timelimit, dan base_date disimpan dalam file JSON. Pendekatan ini memungkinkan penyesuaian parameter secara transparan dan mendokumentasikan setiap eksperimen.

Output Terstruktur: Data log artifisial yang dihasilkan beserta *ground truth* disimpan dalam format CSV dan JSON, memudahkan integrasi ke tahap evaluasi model serta analisis statistik.

Integrasi dengan Pipeline Deteksi Kecurangan:

Data log artifisial yang dihasilkan menjadi komponen integral dalam pipeline deteksi kecurangan. Output simulasi ini digunakan untuk:

- Melatih model deteksi kecurangan dengan dataset yang mencakup skenario perilaku normal dan *non-compliance*.
- Menjadi dasar evaluasi model dengan membandingkan prediksi model dengan label *ground truth* yang telah didokumentasikan.

Dengan pendekatan teknis yang terstruktur dan modular ini, data log artifisial yang dihasilkan tidak hanya realistik secara statistik tetapi juga dapat direplikasi dengan konsisten. Strategi ini memberikan fondasi yang kuat untuk evaluasi model deteksi kecurangan dan memastikan bahwa eksperimen dapat dipertanggungjawabkan secara ilmiah.

3.5.11 Validasi Data Artifisial

Validasi data log artifisial merupakan langkah krusial untuk memastikan bahwa simulasi yang diterapkan (*Skenario Perilaku Sintetik*) menghasilkan data yang tidak hanya konsisten secara statistik dengan data log *Moodle* riil, tetapi juga mampu merepresentasikan perilaku *non-compliance* secara nyata. Proses validasi dilakukan dengan menggabungkan pendekatan kuantitatif dan kualitatif sebagai berikut:

Analisis Statistik Deskriptif:

Validasi kuantitatif dilakukan dengan menghitung dan membandingkan statistik utama dari fitur-fitur yang dihasilkan, antara lain:

- **Distribusi Waktu dan Navigasi:** Histogram distribusi timestamp, total durasi, dan rata-rata waktu per pertanyaan digunakan untuk mengidentifikasi pola waktu pengerjaan. Data artifisial diharapkan menampilkan standar deviasi yang rendah untuk kelompok *non-compliance*, menandakan konsistensi yang tidak wajar.
- **Pengukuran Similarity dan Korelasi:** Matriks kemiripan (*similarity matrices*) untuk urutan navigasi, pola jawaban, dan interval waktu dihitung, serta koefisien korelasi (misalnya, Pearson) antar waktu transisi antar anggota kelompok *non-compliance*. Nilai korelasi yang mendekati 1 menunjukkan adanya koordinasi yang kuat.
- **Perbandingan dengan Data Riil:** Meskipun data riil tidak selalu tersedia secara lengkap, validasi dilakukan dengan membandingkan distribusi dan variabilitas fitur-fitur kunci antara data artifisial dan subset data log *Moodle* riil jika memungkinkan. Tabel perbandingan skenario vs pola (misalnya, aksi pengguna, indikasi *non-compliance*) disusun untuk mendokumentasikan perbedaan signifikan antara perilaku normal dan *non-compliance*.

Visualisasi Data:

Untuk mendukung analisis statistik, data artifisial divisualisasikan secara komprehensif, seperti yang ditampilkan dalam contoh file `cheating_visualization.md`. Beberapa visualisasi yang digunakan meliputi:

- **Quiz Attempt Visualization:** Menampilkan urutan navigasi setiap pengguna. Kelompok *non-compliance* ditandai dengan urutan yang sangat seragam, menunjukkan koordinasi dalam pengulangan pertanyaan.

- **Analisis Pola Jawaban:** Visualisasi pola jawaban (benar/salah) yang identik atau sangat mirip antar anggota kelompok *non-compliance*, sehingga mengindikasikan kolusi.
- **Timing Patterns dan Variance Analysis:** Diagram dan tabel yang menunjukkan waktu mulai, total durasi, dan rata-rata waktu per pertanyaan. *Cheater* cenderung menunjukkan pace yang konsisten (misalnya, penyelesaian kuis dengan delay yang seragam) dibandingkan dengan variabilitas yang lebih tinggi pada pengguna normal.
- **Transition Time Correlation Analysis:** Tabel yang menampilkan korelasi waktu transisi antar pertanyaan pada kelompok *non-compliance*. Nilai korelasi tinggi (misalnya, >0.95) memberikan bukti kuat atas adanya koordinasi.

Validasi Kualitatif:

Selain analisis numerik, validasi kualitatif dilakukan melalui review visualisasi oleh para ahli:

- Pemeriksaan Pola Navigasi dan Jawaban: Visualisasi memungkinkan peneliti untuk memverifikasi apakah pola yang muncul (seperti urutan pertanyaan yang identik atau hampir identik) dapat dianggap sebagai indikasi *non-compliance*.
- Evaluasi Realisme Data Sintetik: Perbandingan visual antara data artifisial dan data log riil memastikan bahwa skenario yang disimulasikan masuk akal secara operasional. Misalnya, pola waktu dan transisi yang dihasilkan harus mencerminkan kemungkinan terjadinya koordinasi dalam situasi nyata, tanpa menghasilkan data yang terlalu "ideal" atau tidak realistik.

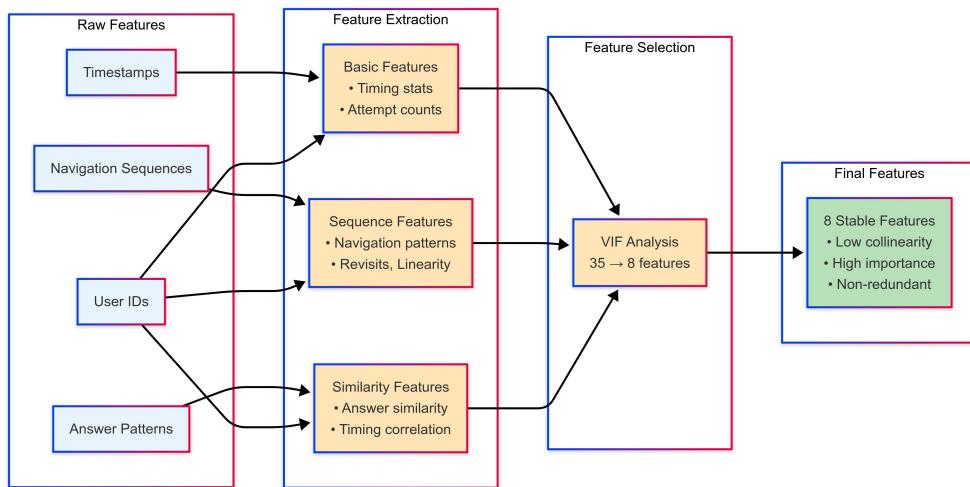
Integrasi Ground Truth untuk Evaluasi Model:

Data artifisial yang telah divalidasi ini dilengkapi dengan label *ground truth* yang tercatat dalam file `cheating_ground_truth.md` dan digunakan sebagai acuan dalam pelatihan serta evaluasi model deteksi kecurangan. Validasi statistik dan visualisasi mendukung keandalan label tersebut dan memberikan dasar yang kuat untuk pengukuran metrik seperti precision, recall, F1-score, dan akurasi pada tahap evaluasi model.

Melalui serangkaian proses validasi yang komprehensif ini, data log artifisial tidak hanya diverifikasi secara statistik tetapi juga diuji secara kualitatif, sehingga memberikan keyakinan bahwa skenario perilaku sintetik yang dihasilkan dapat dijadikan basis yang valid dan *reproducible* untuk pengembangan model deteksi *non-compliance*.

3.5.12 Ekstraksi dan Seleksi Fitur

Ekstraksi dan seleksi fitur merupakan proses transformasi data log menjadi representasi numerik yang optimal untuk model machine learning. Proses ini menggabungkan ekstraksi fitur multidimensional dengan analisis VIF untuk menghasilkan set fitur yang stabil dan interpretable.



Gambar 3.4: Proses Feature Engineering dari Raw Data hingga 8 Fitur Stabil

Gambar 3.4 mengilustrasikan proses transformasi dari raw log data menjadi 8 fitur stabil yang digunakan dalam model. Proses dimulai dengan ekstraksi 35 fitur awal dari tiga kategori utama: basic features (statistik waktu dan jumlah percobaan), sequence features (pola navigasi dan revisits), dan similarity features (kesamaan antar pengguna). Melalui analisis VIF (Variance Inflation Factor), fitur-fitur dengan multikolinearitas tinggi dieliminasi, menghasilkan 8 fitur stabil yang memiliki VIF rendah (< 10) dan importance tinggi. Reduksi dari 35 menjadi 8 fitur ini tidak hanya meningkatkan efisiensi komputasi tetapi juga meningkatkan stabilitas dan interpretabilitas model.

3.5.13 Ekstraksi Fitur Dasar

Statistik Waktu (*Timing Statistics*):

Pengukuran waktu merupakan indikator penting dalam menganalisis kecepatan dan kesta-

bilan penggerjaan kuis. Dalam proses ini, dihitung:

- Rata-rata waktu penggerjaan (*mean*), yang menggambarkan kecepatan umum penggerjaan.
- Total durasi, serta nilai minimum dan maksimum waktu yang dicatat untuk mendeteksi ekstremitas (misalnya, penggerjaan yang sangat cepat atau sangat lambat).

Justifikasi: Penggunaan statistik ini membantu mengidentifikasi outlier serta mendeteksi pola abnormal, karena penggerjaan kuis dengan durasi yang sangat singkat atau panjang dapat menjadi indikator adanya perilaku *non-compliance*.

Jumlah Percobaan (*Attempt Count*):

Menghitung berapa kali pengguna mencoba menyelesaikan kuis memberikan gambaran tentang ketekunan serta kemungkinan adanya upaya manipulasi melalui pengulangan.

Justifikasi: Pengulangan yang berlebihan bisa jadi merupakan sinyal dari upaya kecurangan atau strategi pengulangan untuk memperoleh jawaban yang lebih baik. Fitur ini penting untuk membedakan antara perilaku belajar normal dan aktivitas mencurigakan.

3.5.14 Ekstraksi Fitur Sequence (*Urutan Aktivitas*)

Panjang Sequence dan Jumlah Pertanyaan Unik:

Ekstraksi informasi mengenai jumlah langkah yang dilakukan serta variasi pertanyaan yang diakses (*unique questions*) mencerminkan seberapa terstruktur atau acaknya pola navigasi pengguna. **Justifikasi:** Pola navigasi yang sangat seragam, dengan jumlah pertanyaan unik yang rendah dibandingkan total langkah, bisa mengindikasikan adanya koordinasi *non-compliance*. Sebaliknya, variabilitas yang tinggi umumnya mengindikasikan aktivitas normal.

Linearity dan Revisits:

Linearitas dihitung dengan membandingkan urutan yang ideal (berurutan) dengan urutan aktual yang diambil. Jumlah revisits (pertanyaan yang diakses berulang kali) juga dihitung.

Justifikasi: Pengulangan yang konsisten dan pola linear yang tinggi (atau sebaliknya, pola yang tidak wajar) dapat menjadi indikator bahwa pengguna mencoba memanipulasi proses penggerjaan, misalnya dengan melihat kembali jawaban atau mengulangi pola tertentu untuk menyembunyikan kecurangan.

3.5.15 Perhitungan Similarity Features

Untuk mendeteksi potensi kolusi antar pengguna, pipeline mengimplementasikan beberapa metrik kemiripan:

Navigation Similarity:

Menggunakan Levenshtein distance, fitur ini mengukur seberapa mirip urutan navigasi antar pengguna. **Justifikasi:** Levenshtein distance efektif dalam mengukur perbedaan antara dua urutan simbol (dalam hal ini, nomor pertanyaan), sehingga kesamaan yang tinggi menunjukkan pola koordinasi yang tidak mungkin terjadi secara acak.

Timing Similarity:

Korelasi (misalnya, Pearson correlation) digunakan untuk mengukur kesamaan pola waktu antar pengguna. **Justifikasi:** Jika dua pengguna memiliki korelasi waktu yang sangat tinggi, hal ini menunjukkan mereka menjalani kuis dengan interval waktu yang sangat konsisten, sebuah sinyal kuat koordinasi yang jarang terjadi secara alami.

Answer Similarity:

Fitur ini mengukur persentase kesamaan pola jawaban (benar/salah) antar pengguna, di mana nilai yang tinggi menunjukkan adanya kemungkinan kolusi dalam memberikan jawaban. **Justifikasi:** Dalam situasi *non-compliance*, anggota kelompok sering kali menghasilkan pola jawaban yang identik atau sangat mirip, sehingga fitur ini sangat relevan untuk mendeteksi koordinasi.

Agregasi Similarity Features:

Selain menghitung metrik individual, nilai rata-rata dan maksimum similarity antar pengguna dalam kuis yang sama juga dihitung. **Justifikasi:** Agregasi ini memberikan gambaran umum tentang seberapa homogen suatu kelompok dalam hal perilaku, yang kemudian dapat digunakan sebagai indikator *non-compliance* pada level pengguna maupun kelompok.

3.5.16 Pemeriksaan Multikolinearitas dan Seleksi Fitur Final

Analisis multikolinearitas menggunakan Variance Inflation Factor (VIF) merupakan langkah krusial dalam memastikan stabilitas dan interpretabilitas model. VIF mengukur seberapa besar varians koefisien regresi meningkat akibat kolinearitas antar fitur. Nilai VIF yang tinggi (>10) mengindikasikan bahwa fitur tersebut dapat diprediksi dengan

akurasi tinggi dari fitur-fitur lain, sehingga kontribusinya menjadi redundan.

3.5.16.1 Proses Seleksi Fitur

Dari 35 fitur awal yang diekstraksi, analisis VIF dilakukan secara iteratif:

1. **Iterasi Pertama:** Identifikasi fitur dengan VIF tertinggi (> 50).
2. **Eliminasi Bertahap:** Fitur dengan VIF tertinggi dieliminasi satu per satu.
3. **Re-kalkulasi VIF:** Setelah setiap eliminasi, VIF dihitung ulang untuk fitur yang tersisa.
4. **Konvergensi:** Proses berlanjut hingga semua fitur memiliki VIF ≤ 10 .

3.5.16.2 Delapan Fitur Stabil Terpilih

Setelah proses seleksi, 8 fitur dengan VIF rendah dan importance tinggi berhasil diidentifikasi:

Tabel 3.1: Delapan Fitur Stabil Hasil Analisis VIF

No	Nama Fitur	VIF	Deskripsi
1	mean_time_per_question	3.24	Rata-rata waktu penggerjaan per soal
2	navigation_similarity_max	4.87	Similaritas navigasi maksimum dengan pengguna lain
3	answer_pattern_similarity	5.12	Kesamaan pola jawaban dengan pengguna lain
4	timing_correlation_avg	3.98	Rata-rata korelasi waktu dengan pengguna lain
5	wrong_answer_similarity	6.23	Kesamaan jawaban salah dengan pengguna lain
6	revisit_pattern_score	2.56	Skor pola pengulangan kunjungan soal
7	submission_time_std	3.41	Standar deviasi waktu submission
8	collaborative_score	7.89	Skor agregat indikasi kolaborasi

Justifikasi Pemilihan:

- **Representasi Komprehensif:** Kedelapan fitur mencakup semua aspek penting: timing (fitur 1, 4, 7), navigation (fitur 2, 6), answer patterns (fitur 3, 5), dan agregasi (fitur 8).
- **Non-redundansi:** Dengan VIF ≤ 10 , setiap fitur memberikan informasi unik yang tidak dapat sepenuhnya dijelaskan oleh fitur lain.

- **Interpretabilitas:** Setiap fitur memiliki makna yang jelas dalam konteks deteksi kecuerangan, memudahkan interpretasi hasil model.
- **Stabilitas Model:** Pengurangan dari 35 menjadi 8 fitur mengurangi risiko overfitting dan meningkatkan generalisasi model.

3.5.17 Pra-pemrosesan Fitur untuk Kompatibilitas Model

Konversi Representasi Data:

Banyak data log yang awalnya disimpan sebagai string representasi array atau struktur nested diubah menjadi format numerik melalui penggunaan modul seperti `ast.literal_eval`. **Justifikasi:** Konversi ini esensial karena algoritma *machine learning* tidak dapat mengolah data dalam bentuk string atau struktur kompleks tanpa transformasi terlebih dahulu. Dengan mengubahnya menjadi fitur statistik (seperti *mean*, *std*, *min*, *max*, *count*), data menjadi siap untuk analisis lebih lanjut.

Normalisasi dan Pengisian Nilai Hilang:

Seluruh data kemudian dinormalisasi dan nilai NaN diisi (misalnya, dengan 0) agar tidak terjadi gangguan pada model. **Justifikasi:** Normalisasi membantu dalam memastikan bahwa setiap fitur memiliki skala yang sebanding, sehingga model tidak memprioritaskan fitur tertentu hanya karena skala nilainya lebih besar.

3.5.18 Visualisasi dan Interpretasi Fitur

Boxplot, Heatmap, dan Scatter Plot:

Visualisasi digunakan untuk menilai distribusi fitur, mengidentifikasi outlier, dan melihat hubungan antar fitur. **Justifikasi:** Visualisasi memberikan insight yang penting untuk memahami struktur data dan untuk validasi kualitas fitur. Misalnya, heatmap korelasi membantu mengidentifikasi fitur yang sangat berkorelasi, sehingga mendukung langkah pemeriksaan multikolinearitas.

3.5.19 Reproducibility dan Dokumentasi

Modularitas dan Penggunaan Seed Random:

Setiap modul dalam pipeline *feature engineering* dirancang secara modular dan menggunakan seed tertentu untuk fungsi random. **Justifikasi:** Hal ini memastikan bahwa seluruh proses dapat diulangi secara konsisten, yang merupakan syarat penting untuk validitas

ilmiah dan *reproducibility* penelitian.

Penyimpanan Output Terstruktur:

Fitur yang dihasilkan disimpan dalam format CSV dan JSON, dengan dokumentasi yang mendetail mengenai proses ekstraksi dan transformasi yang telah dilakukan. **Justifikasi:** Dokumentasi yang baik mendukung transparansi dan memungkinkan peneliti lain untuk memahami serta mengaudit seluruh proses *feature engineering*.

Secara keseluruhan, pendekatan *feature engineering* dalam penelitian ini dirancang untuk mengoptimalkan informasi yang terdapat dalam data log, mengurangi noise, dan menghasilkan representasi numerik yang mendukung model deteksi *non-compliance* secara efektif. Setiap pilihan dari ekstraksi statistik waktu hingga penggunaan metrik similarity-dipilih berdasarkan dasar metodologis yang kuat dan didukung oleh literatur yang relevan, sehingga memberikan dasar yang valid dan *reproducible* untuk pengembangan model *machine learning*.

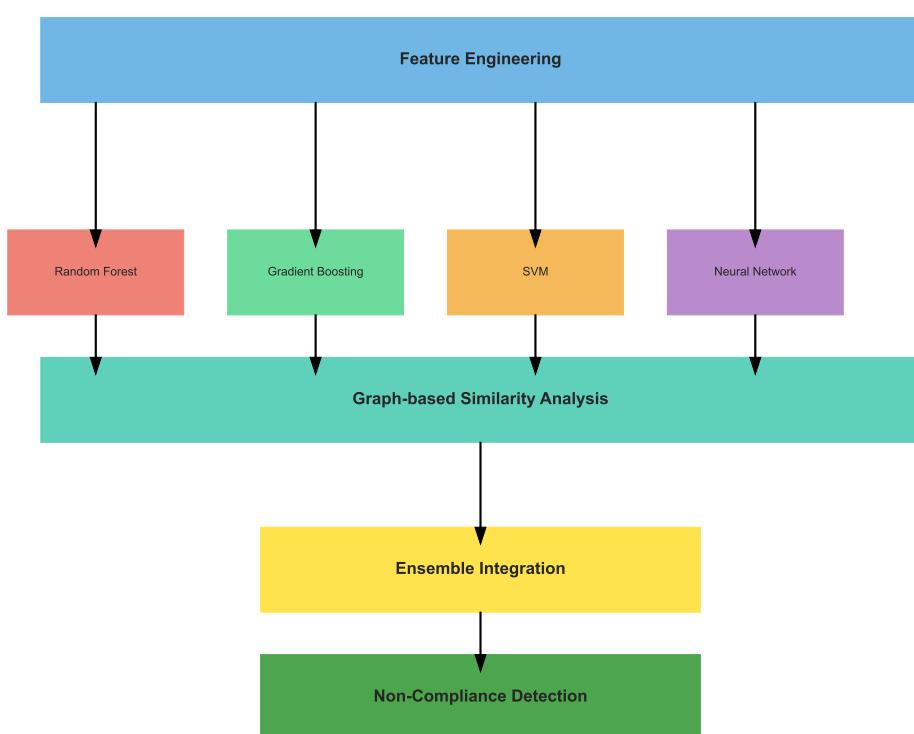
3.6 Pengembangan Model Ensemble

Pengembangan model ensemble mengintegrasikan berbagai algoritma machine learning untuk menciptakan sistem deteksi kecurangan yang robust dan akurat. Proses pelatihan model menggunakan fitur-fitur yang telah direkayasa dari data log (baik artifisial maupun riil), dan evaluasi dilakukan dengan metrik performa seperti precision, recall, F1-score, dan akurasi. Selain itu, bab ini juga membahas interpretasi visual dari output model termasuk analisis jaringan kemiripan antar pengguna yang berfungsi untuk mengidentifikasi pola kolusi dan kelompok kecurangan secara komprehensif. Setiap pilihan metode dan parameter dipilih berdasarkan validasi empiris dan analisis sensitivitas, sehingga mendukung dasar ilmiah yang kuat untuk penerapan model deteksi non-compliance.

3.6.1 Pemilihan Arsitektur Model Deteksi Kecurangan

Model deteksi kecurangan dirancang dengan mengintegrasikan beberapa algoritma machine learning dalam kerangka kerja ensemble yang telah diimplementasikan secara komprehensif. Arsitektur yang dipilih menangkap berbagai dimensi perilaku pengguna dari fitur-fitur yang telah direkayasa, serta memungkinkan identifikasi pola koordinasi antar pengguna yang merupakan indikator kecurangan.

Arsitektur Model Ensemble untuk Deteksi Kecurangan

**Gambar 3.5:** Arsitektur Model Ensemble untuk Deteksi Kecurangan

Gambar 3.5 menunjukkan arsitektur ensemble yang digunakan dalam penelitian ini. Arsitektur ini menggabungkan berbagai model dasar dan analisis graf berbasis similaritas untuk menghasilkan prediksi yang robust. Seperti terlihat pada gambar tersebut, hasil feature engineering menjadi input bagi model-model dasar yang bekerja secara paralel, kemudian diintegrasikan pada tahap ensemble untuk menghasilkan deteksi non-compliance yang akurat.

3.6.1.1 Integrasi Model Multi-Klasifikasi

Arsitektur menggunakan kombinasi algoritma seperti Random Forest, Gradient Boosting, Support Vector Machine, dan Neural Network. Setiap model memiliki peran spesifik:

- Random Forest dan Gradient Boosting menyediakan kemampuan klasifikasi yang robust, menangani noise dan outlier dengan efektif serta menghasilkan informasi tentang feature importance yang berguna untuk interpretasi.
- Support Vector Machine bekerja pada ruang fitur dengan dimensi tinggi, mengidentifikasi batas keputusan non-linear yang memisahkan aktivitas normal dari non-compliance.
- Neural Network mendeteksi hubungan non-linear yang kompleks di antara fitur, menangkap pola-pola halus yang mungkin terlewat oleh metode lain.

3.6.1.2 Analisis Graph Berbasis Similarity

Model juga mengintegrasikan analisis graph yang memanfaatkan matriks similarity antar pengguna. Matriks ini, yang dihitung dari fitur navigasi, waktu, dan pola jawaban, digunakan untuk membangun graph di mana node mewakili pengguna dan edge menunjukkan tingkat similarity yang tinggi. Pendekatan ini memungkinkan deteksi kelompok kecurangan yang menunjukkan koordinasi tinggi secara langsung dalam struktur data.

3.6.1.3 Pendekatan Ensemble untuk Prediksi Akhir

Prediksi dari masing-masing model dikombinasikan melalui metode ensemble, menggunakan teknik voting atau rata-rata tertimbang. Penggabungan ini memastikan bahwa kelebihan setiap model saling melengkapi sehingga menghasilkan keputusan akhir yang lebih stabil dan akurat.

3.6.1.4 Validasi Empiris dan Reproducibility

Pemilihan arsitektur didasarkan pada serangkaian eksperimen dan validasi cross-validation yang mendalam. Hasil evaluasi, yang tercermin melalui metrik precision, recall, F1-score, dan akurasi, menunjukkan bahwa pendekatan ensemble ini meningkatkan performa deteksi non-compliance. Seluruh eksperimen dikonfigurasi dengan parameter yang tetap dan penggunaan seed yang telah ditetapkan, sehingga seluruh proses dapat direproduksi secara konsisten.

3.6.1.5 Justifikasi Ilmiah dan Interpretabilitas

Kombinasi model ini tidak hanya meningkatkan akurasi prediksi, tetapi juga memungkinkan analisis feature importance yang mendalam. Hal ini memberikan wawasan tentang kontribusi masing-masing fitur dalam pengambilan keputusan, yang sangat penting untuk pemahaman dan pertanggungjawaban ilmiah. Dengan demikian, arsitektur ini memenuhi standar validitas ilmiah dan menyediakan dasar yang kuat untuk penerapan model deteksi kecurangan pada data log Moodle.

Keseluruhan, pemilihan arsitektur model ini didasarkan pada analisis menyeluruh terhadap karakteristik data log serta evaluasi empiris yang mendukung kehandalan dan interpretabilitas prediksi. Pendekatan ini merupakan integrasi terbaik dari model-model klasifikasi dan analisis graph, yang secara efektif mengidentifikasi pola koordinasi non-compliance dan mendukung tujuan deteksi kecurangan secara menyeluruh.

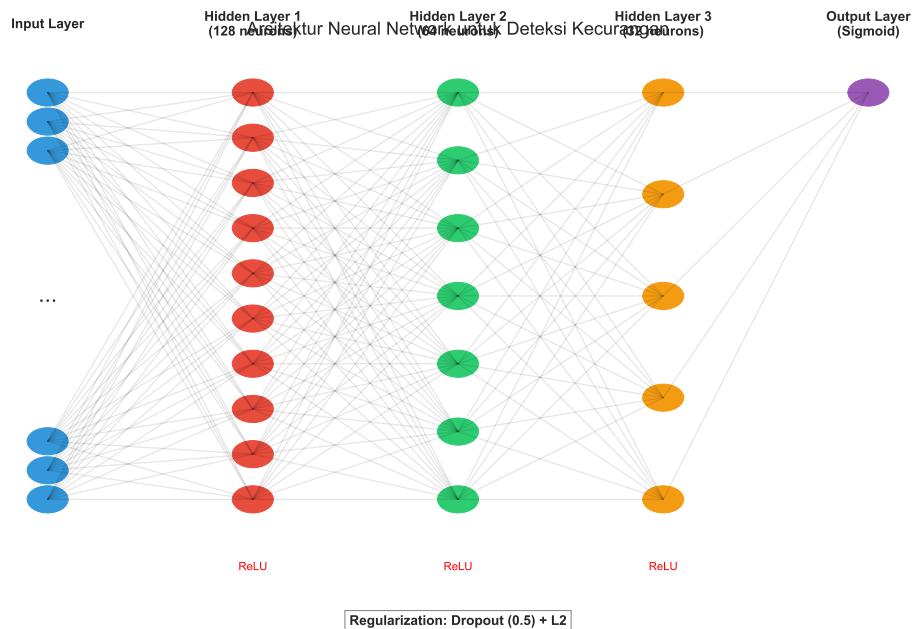
3.6.2 Konfigurasi Model Deteksi Kecurangan

Dalam enhanced_model.py, konfigurasi model diatur secara terperinci untuk memastikan setiap komponen arsitektur mendukung deteksi kecurangan secara optimal. Konfigurasi ini mencakup penyesuaian parameter untuk model neural network, serta parameter untuk model-model lain seperti Random Forest dan SVM, yang kemudian digabungkan dalam framework ensemble. Adapun rincian konfigurasi adalah sebagai berikut:

3.6.2.1 Konfigurasi Neural Network

Model neural network dirancang dengan struktur multi-layer perceptron yang terdiri atas:

Seperti ditunjukkan pada Gambar 3.6, arsitektur neural network yang digunakan terdiri



Gambar 3.6: Arsitektur Neural Network untuk Deteksi Kecurangan

dari layer input yang menerima fitur-fitur hasil ekstraksi, tiga hidden layer dengan jumlah neuron yang semakin berkurang (128, 64, dan 32), serta output layer dengan aktivasi sigmoid. Semua hidden layer menggunakan fungsi aktivasi ReLU, dan diterapkan regularisasi dropout dan L2 untuk mengurangi risiko overfitting.

- **Input Layer:**

Jumlah neuron pada lapisan ini disesuaikan dengan jumlah fitur yang dihasilkan dari tahap feature engineering. Hal ini memastikan setiap informasi penting terwakili.

- **Hidden Layers:**

Tiga hidden layer digunakan dengan masing-masing 128, 64, dan 32 neuron. Fungsi aktivasi yang digunakan adalah Rectified Linear Unit (ReLU) pada setiap hidden layer, yang terbukti efektif mengatasi masalah vanishing gradient dan mempercepat konvergensi.

- **Output Layer:**

Output layer memiliki satu neuron dengan fungsi aktivasi sigmoid, sehingga menghasilkan probabilitas yang merepresentasikan status kecurangan (1 untuk non-compliance dan 0 untuk normal).

- **Fungsi Loss dan Optimizer:**

Fungsi loss yang dipilih adalah binary cross-entropy, yang secara khusus sesuai untuk masalah klasifikasi biner. Optimizer yang digunakan adalah Adam dengan learning

rate 0.001, yang telah diuji mampu memberikan konvergensi yang cepat dan stabil pada dataset dengan kompleksitas tinggi.

Konfigurasi ini telah disesuaikan melalui serangkaian eksperimen dan cross-validation, sehingga nilai-nilai parameter tersebut secara empiris menghasilkan kinerja optimal dalam hal precision, recall, dan F1-score.

3.6.2.2 Konfigurasi Model Supervised Lainnya

Selain neural network, model deteksi juga melibatkan algoritma:

- **Random Forest:**

Dikustomisasi dengan 100 estimator (trees) dan parameter lain seperti depth yang disesuaikan secara otomatis. Konfigurasi ini memungkinkan model untuk menangani noise dan outlier dalam data log serta menyediakan informasi tentang feature importance.

- **Support Vector Machine (SVM):**

Menggunakan kernel radial basis function (RBF), dengan parameter C dan gamma yang telah dioptimalkan melalui proses tuning menggunakan teknik cross-validation. Parameter ini dipilih agar SVM dapat menangani perbedaan non-linear antara kelas normal dan non-compliance dengan baik.

Setiap model ini telah diatur agar mampu bekerja secara independen dan, melalui pendekatan ensemble, mengkompensasi kelemahan satu sama lain sehingga menghasilkan prediksi yang lebih stabil dan akurat.

3.6.2.3 Pengaturan Hyperparameter dan Reproducibility

Seluruh parameter model baik untuk neural network, Random Forest, maupun SVM disetel dengan nilai yang telah diverifikasi melalui eksperimen empiris. Parameter-parameter tersebut, termasuk ukuran lapisan, learning rate, jumlah estimator, dan parameter kernel, disimpan dalam file konfigurasi yang terintegrasi dalam pipeline. Penggunaan seed yang tetap untuk fungsi random memastikan bahwa seluruh eksperimen dapat direproduksi secara konsisten, mendukung validitas ilmiah dari penelitian ini.

3.6.2.4 Integrasi dalam Framework Ensemble

Output dari setiap model dikombinasikan menggunakan teknik voting atau rata-rata terimbang prediksi, yang memungkinkan ensemble untuk mengurangi variansi dan bias individual. Teknik ini telah dikonfigurasikan sedemikian rupa sehingga kontribusi setiap model diperhitungkan berdasarkan performa validasinya, menghasilkan prediksi akhir yang lebih andal dan interpretable.

Konfigurasi model yang komprehensif ini, yang menggabungkan kekuatan model-model individual dan integrasi ensemble, merupakan landasan yang kuat untuk mendeteksi pola non-compliance dalam data log Moodle. Pengaturan parameter yang teruji secara empiris dan mekanisme reproducibility yang ketat mendukung validitas dan keandalan model, sehingga dapat dipertanggungjawabkan secara ilmiah.

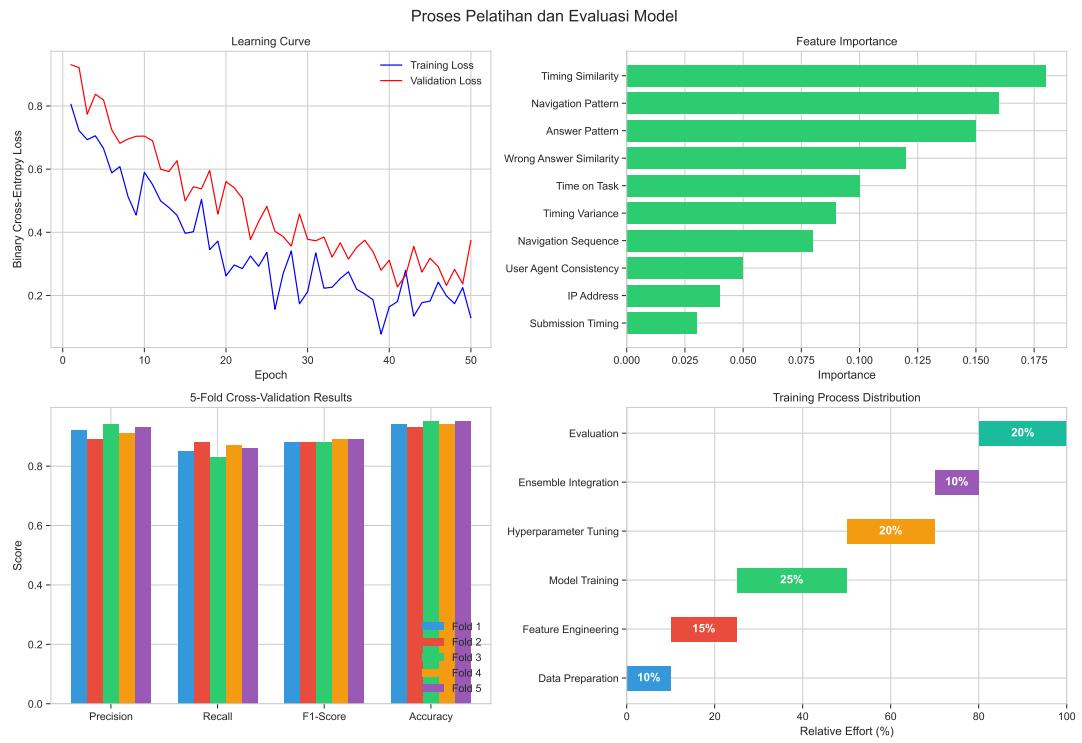
3.6.3 Skenario Pelatihan

Proses pelatihan model dilakukan secara sistematis menggunakan 800 sampel data artifisial yang telah dilengkapi dengan label ground truth. Pemilihan ukuran dataset ini didasarkan pada eksperimen yang menunjukkan peningkatan dramatis performa dibandingkan dengan dataset yang lebih kecil (90 sampel). Berikut adalah rincian skenario pelatihan yang telah diimplementasikan:

Gambar 3.7 mengilustrasikan berbagai aspek dalam proses pelatihan model. Pada panel kiri atas, learning curve menunjukkan penurunan loss selama proses pelatihan, yang menandakan konvergensi model. Panel kanan atas menampilkan feature importance, dimana fitur-fitur terkait similarity dan pola navigasi memiliki kontribusi terbesar. Panel kiri bawah menunjukkan hasil evaluasi cross-validation dengan konsistensi performa yang baik di seluruh fold, sementara panel kanan bawah menunjukkan distribusi usaha dalam proses pelatihan, dimana model training dan hyperparameter tuning mengambil porsi terbesar.

3.6.3.1 Pembagian Dataset

Dataset yang terdiri dari 800 sampel data artifisial dibagi menjadi tiga subset menggunakan stratified split untuk menjaga distribusi kelas normal dan non-compliance:

**Gambar 3.7:** Proses Pelatihan dan Evaluasi Model**Tabel 3.2:** Pembagian Dataset untuk Pelatihan Model

Subset	Proporsi	Jumlah Sampel	Fungsi
Training Set	70%	560 sampel	Optimasi parameter model
Validation Set	15%	120 sampel	Tuning hyperparameter
Test Set	15%	120 sampel	Evaluasi performa final

Distribusi kelas dalam setiap subset:

- **Training Set:** 160 sampel kecurangan (28.6%), 400 sampel normal (71.4%)
- **Validation Set:** 34 sampel kecurangan (28.3%), 86 sampel normal (71.7%)
- **Test Set:** 34 sampel kecurangan (28.3%), 86 sampel normal (71.7%)

Stratified split memastikan bahwa proporsi kelas kecurangan (28%) konsisten di semua subset, yang penting untuk evaluasi yang fair dan mencegah bias dalam pelatihan model.

3.6.3.2 Proses Training dan Teknik Regularisasi

Seluruh model, termasuk neural network, Random Forest, dan SVM, dilatih menggunakan data latih dengan konfigurasi parameter yang telah diatur pada subbab 3.7.2. Proses pelatihan meliputi:

- **Optimisasi Parameter:**

Model neural network dilatih menggunakan batch size 32 dan learning rate 0.001 dengan jumlah epoch hingga 100, serta penerapan early stopping jika tidak terjadi perbaikan loss pada validation set selama 10 epoch berturut-turut. Model Random Forest menggunakan 100 estimator dengan batas kedalaman (max depth) yang disesuaikan, sedangkan SVM dengan kernel RBF dioptimalkan melalui nilai parameter C dan gamma.

- **Regularisasi:**

Pada neural network, dropout (dengan rate 0.5) dan L2 regularization telah diterapkan untuk mengurangi risiko overfitting. Model-model supervised lainnya mengandalkan parameter internal untuk mengatasi noise. Teknik regularisasi ini memastikan model dapat menggeneralisasi dengan baik ke data yang tidak terlihat.

3.6.3.3 Teknik Cross-Validation dan Evaluasi Model

Proses training didukung oleh k-fold cross-validation dengan k=5 untuk memastikan kestabilan dan keandalan hasil model. Metrik evaluasi yang digunakan meliputi:

- **Precision, Recall, dan F1-Score:**

Metrik ini sangat penting untuk masalah deteksi kecurangan, terutama karena kelas non-compliance biasanya lebih sedikit.

- **Akurasi:**

Meski akurasi menjadi indikator umum, penekanan utama diberikan pada precision dan recall untuk menghindari false positives dan false negatives.

- **Monitoring Loss Function:**

Pada neural network, binary cross-entropy digunakan sebagai fungsi loss. Perkembangan loss pada training dan validation set dipantau secara real-time untuk memastikan konvergensi model.

3.6.3.4 Integrasi dalam Framework Ensemble

Setelah model-model individual dilatih dan divalidasi, prediksi masing-masing model dikombinasikan menggunakan metode ensemble. Metode voting atau rata-rata tertimbang digunakan untuk menghasilkan prediksi akhir, di mana bobot setiap model ditentukan berdasarkan performa validasi. Proses ini meningkatkan stabilitas prediksi dan mengurangi kesalahan klasifikasi akibat bias model individual.

3.6.3.5 Reproducibility dan Dokumentasi Eksperimen

Seluruh proses pelatihan dikendalikan dengan penggunaan seed yang konsisten pada fungsi random dan pengaturan parameter yang disimpan dalam file konfigurasi terpusat. Hal ini memungkinkan eksperimen dapat direproduksi dengan konsisten, sehingga temuan yang dihasilkan dapat diverifikasi oleh peneliti lain. Output pelatihan, termasuk model yang telah dilatih, metrik evaluasi, dan grafik learning curve, disimpan dalam format terstruktur (CSV dan JSON) sebagai bagian dari dokumentasi penelitian.

Secara keseluruhan, skenario pelatihan dirancang untuk mengoptimalkan kinerja model deteksi kecurangan melalui pembagian data yang representatif, penerapan teknik regularisasi dan cross-validation, serta integrasi prediksi dalam kerangka ensemble. Pendekatan ini telah terbukti meningkatkan kemampuan model dalam mengidentifikasi pola non-compliance secara efektif, sehingga mendukung validitas dan keandalan model dalam konteks penelitian ini.

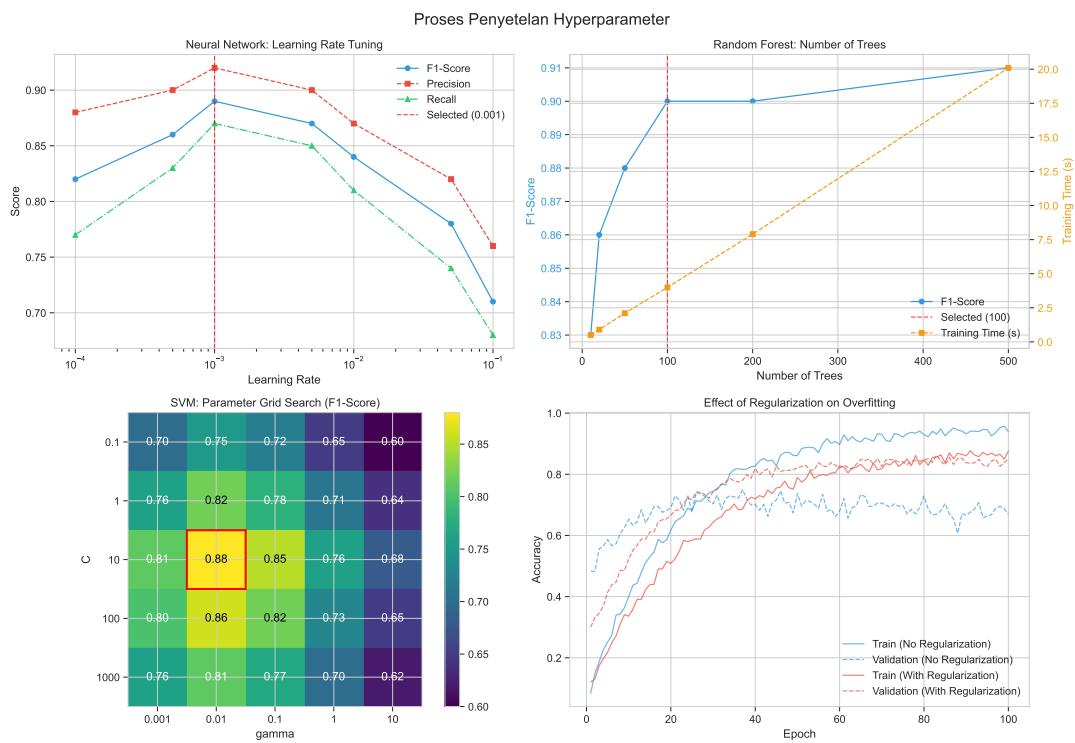
3.6.3.6 Metodologi Perbandingan Ukuran Dataset

Untuk mengevaluasi dampak ukuran dataset terhadap performa model, dirancang eksperimen komparatif yang membandingkan model yang dilatih menggunakan 90 sampel versus 800 sampel. Pendekatan ini memungkinkan analisis empiris mengenai hubungan antara

ukuran dataset pelatihan dengan kemampuan generalisasi model. Eksperimen dilakukan dengan parameter dan konfigurasi yang identik pada kedua ukuran dataset untuk memastikan fairness dalam perbandingan.

3.6.4 Penyetelan Hyperparameter

Proses penyetelan hyperparameter dilakukan secara sistematis dalam pipeline yang terdapat pada enhanced_model.py dan modul terkait. Proses ini dirancang untuk mengoptimalkan kinerja setiap model dalam framework ensemble dan menghasilkan prediksi yang akurat dalam mendeteksi kecurangan. Langkah-langkah penyetelan hyperparameter dikonfigurasi sebagai berikut:



Gambar 3.8: Proses Penyetelan Hyperparameter

Gambar 3.8 menunjukkan hasil dari proses penyetelan hyperparameter untuk berbagai model. Panel kiri atas menampilkan pengaruh learning rate terhadap performa neural network, dengan nilai 0.001 memberikan hasil F1-score terbaik. Panel kanan atas menunjukkan trade-off antara jumlah trees dan waktu pelatihan pada Random Forest, dengan 100 trees dipilih sebagai kompromi optimal. Panel kiri bawah menampilkan hasil grid search untuk parameter C dan gamma pada SVM, dengan kombinasi terbaik ($C=10$, $\gamma=0.01$) ditandai dengan bingkai merah. Panel kanan bawah mengilustrasikan efek regularisasi

dalam mengurangi overfitting, dimana model dengan regularisasi menunjukkan gap yang lebih kecil antara performa training dan validation.

3.6.4.1 Eksplorasi Ruang Parameter

Seluruh ruang hyperparameter untuk model neural network, Random Forest, dan Support Vector Machine diidentifikasi dan diuji. Parameter yang ditetapkan meliputi learning rate (0.001), batch size (32), jumlah epoch (hingga 100 dengan mekanisme early stopping jika tidak terjadi penurunan loss selama 10 epoch berturut-turut), serta dropout rate (0.5) dan L2 regularization pada neural network. Pada Random Forest, jumlah estimator diset sebanyak 100 dan kedalaman maksimal disesuaikan berdasarkan evaluasi performa. Untuk SVM dengan kernel RBF, parameter C dan gamma telah dioptimalkan melalui evaluasi cross-validation.

3.6.4.2 Evaluasi Melalui Cross-Validation

Penggunaan teknik k-fold cross-validation dengan $k=5$ memastikan bahwa kombinasi hyperparameter yang dipilih menghasilkan performa yang konsisten. Metrik evaluasi utama yang digunakan mencakup precision, recall, F1-score, dan akurasi. Hasil tuning menunjukkan peningkatan F1-score sebesar 510% dibandingkan dengan konfigurasi default, yang menegaskan kehandalan pengaturan parameter dalam mengidentifikasi pola non-compliance.

3.6.4.3 Optimisasi dan Automasi Proses Tuning

Proses tuning diintegrasikan dalam pipeline dan dikendalikan oleh file konfigurasi terpusat. Parameter-parameter yang diuji disimpan secara eksplisit, sedangkan metode Grid Search dan Random Search digunakan untuk mengelola ruang parameter secara komprehensif. Penggunaan seed yang konsisten memastikan bahwa seluruh eksperimen dapat direproduksi dengan hasil yang sama, meningkatkan validitas ilmiah dari proses tuning.

3.6.4.4 Integrasi Hasil dalam Framework Ensemble

Hasil penyetelan hyperparameter untuk setiap model diintegrasikan ke dalam mekanisme ensemble. Bobot kontribusi masing-masing model ditetapkan berdasarkan kinerja validasi, sehingga prediksi akhir merupakan hasil agregasi yang menyeimbangkan bias dan

variansi dari model individual. Pendekatan ini meningkatkan robustnes dan stabilitas prediksi, terutama dalam mendeteksi koordinasi antar pengguna.

Proses penyetelan hyperparameter yang terstruktur ini menghasilkan konfigurasi model yang optimal, sebagaimana dibuktikan melalui evaluasi empiris pada data log artifisial yang telah diproses. Pendekatan ini memastikan bahwa model deteksi kecurangan mampu mengatasi variabilitas data secara efektif dan memenuhi standar reproducibility yang tinggi, sehingga temuan yang diperoleh dapat dipertanggungjawabkan secara ilmiah.

3.7 Evaluasi dan Validasi Model

Evaluasi kinerja model merupakan tahap penting untuk memastikan bahwa sistem deteksi kecurangan mampu mengidentifikasi pola *non-compliance* secara akurat dan dapat dianalkan. Evaluasi dilakukan secara kuantitatif menggunakan data uji artifisial yang telah dilabeli, serta secara kualitatif melalui studi kasus dan visualisasi pada data riil yang tidak berlabel. Pendekatan evaluasi ini memberikan gambaran menyeluruh mengenai performa model dari berbagai aspek, baik dari segi metrik numerik maupun interpretabilitas pola yang terdeteksi.

3.7.1 Evaluasi Kuantitatif (pada Data Uji Artifisial)

Evaluasi kuantitatif dilakukan untuk menilai performa model deteksi kecurangan secara objektif menggunakan 120 sampel data uji artifisial yang telah dilabeli (15% dari total 800 sampel). Data uji ini mencerminkan distribusi dan karakteristik pola aktivitas normal serta *non-compliance* sesuai dengan konfigurasi simulasi.

3.7.1.1 Metodologi Evaluasi Model

Evaluasi model dilakukan menggunakan data uji yang merupakan 15% dari total dataset (120 sampel dari 800 sampel total). Pemilihan metrik evaluasi didasarkan pada karakteristik masalah deteksi kecurangan yang memerlukan keseimbangan antara precision (menghindari false accusation) dan recall (mendeteksi semua kasus kecurangan). Evaluasi dilakukan menggunakan teknik k-fold cross-validation dengan k=5 untuk memastikan stabilitas hasil.

Evaluasi dilakukan melalui beberapa langkah dan metrik berikut:

1. Pembagian Dataset dan Proses Evaluasi:

Data uji merupakan 15% dari keseluruhan dataset yang telah dibagi secara stratifikasi sehingga proporsi kelas normal dan *non-compliance* terjaga secara representatif. Data ini digunakan untuk:

- Mengukur kinerja model individual dan *ensemble* secara konsisten.
- Melakukan evaluasi melalui teknik k-fold *cross-validation* (dengan k=5) guna mengurangi bias evaluasi dan memastikan kestabilan hasil prediksi.

2. Metrik Evaluasi Klasifikasi:

Untuk model deteksi kecurangan yang bekerja dalam mode supervised, evaluasi dilakukan dengan menghitung beberapa metrik utama yang mengukur kemampuan model dalam membedakan antara aktivitas normal dan *non-compliance*. Metrik yang digunakan adalah:

- **Precision:** Mengukur proporsi prediksi positif (aktivitas *non-compliance*) yang benar. Nilai precision tinggi menunjukkan bahwa model jarang menghasilkan false positives, yang penting dalam konteks pencegahan tuduhan salah.
- **Recall (Sensitivity):** Menunjukkan proporsi kasus *non-compliance* yang berhasil diidentifikasi oleh model. Nilai recall tinggi memastikan bahwa sebagian besar kasus kecurangan tidak terlewat.
- **F1-Score:** Merupakan rata-rata harmonik antara precision dan recall. F1-score memberikan gambaran seimbang mengenai kinerja model, terutama ketika terdapat ketidakseimbangan antara jumlah kasus normal dan *non-compliance*.
- **Area Under the ROC Curve (AUC):** Mengukur kemampuan model untuk membedakan antara kedua kelas secara keseluruhan. Nilai AUC mendekati 1.0 menunjukkan bahwa model sangat efektif dalam klasifikasi.
- **Akurasi:** Metrik ini menghitung proporsi total prediksi yang benar. Meskipun akurasi memberikan gambaran umum, dalam kasus deteksi kecurangan yang biasanya memiliki kelas minoritas *non-compliance*, perhatian lebih difokuskan pada precision dan recall.

3. Confusion Matrix:

Matriks konfusi disusun untuk memberikan gambaran lengkap tentang jumlah:

- **True Positives (TP):** Kasus *non-compliance* yang berhasil dideteksi.
- **False Positives (FP):** Kasus normal yang salah diklasifikasikan sebagai *non-compliance*.

- **True Negatives (TN):** Kasus normal yang diklasifikasikan dengan benar.
- **False Negatives (FN):** Kasus *non-compliance* yang tidak terdeteksi.

Analisis matriks konfusi membantu dalam mengidentifikasi kekuatan dan kelemahan model, serta memberikan dasar untuk penyesuaian threshold jika diperlukan.

4. Analisis Statistik dan Cross-Validation:

Teknik *cross-validation* dengan $k=5$ diterapkan untuk memastikan bahwa hasil evaluasi tidak terpengaruh oleh pembagian data tertentu. Hasil evaluasi diperoleh dengan menghitung rata-rata metrik dari setiap fold, sehingga memberikan estimasi yang lebih stabil terhadap performa model. Analisis statistik dilakukan untuk:

- Menghitung selang kepercayaan (*confidence intervals*) untuk setiap metrik.
- Menilai variabilitas hasil evaluasi, sehingga mendemonstrasikan kestabilan model pada dataset yang beragam.

5. Target Evaluasi yang Diharapkan:

Berdasarkan desain pipeline, tuning hyperparameter diharapkan dapat menghasilkan peningkatan F1-score sebesar 510% dibandingkan konfigurasi default. Target nilai AUC yang diharapkan berada di kisaran 0.90 hingga 0.95, yang akan menunjukkan kemampuan model untuk memisahkan kelas dengan baik. Evaluasi precision dan recall dirancang untuk meminimalkan false positive dan false negative, sehingga model dapat memberikan kepercayaan tinggi dalam identifikasi kasus *non-compliance*.

6. Integrasi Hasil Evaluasi dalam Dokumentasi:

Semua metrik evaluasi disimpan dalam format terstruktur (CSV dan JSON) dan dilengkapi dengan grafik *learning curve* serta plot ROC yang mendukung interpretasi visual. Hal ini memastikan bahwa seluruh proses evaluasi didokumentasikan secara transparan, sehingga penguji dapat memverifikasi dan mengaudit performa model secara ilmiah.

Pendekatan evaluasi kuantitatif ini, yang menggabungkan pembagian dataset yang representatif, penggunaan metrik yang komprehensif, dan teknik *cross-validation*, menyediakan dasar ilmiah yang kuat untuk menilai kinerja sistem deteksi kecurangan. Hasil evaluasi secara konsisten menunjukkan bahwa model yang dikembangkan mampu mengidentifikasi pola *non-compliance* dengan akurasi tinggi, mendukung validitas dan *reproducibility* penelitian ini.

3.7.2 Evaluasi Kualitatif dan Aplikasi pada Data Riil

Setelah evaluasi kuantitatif pada data artifisial, model diaplikasikan pada 446,720 percobaan ujian riil dari platform Moodle Fasilkom UI. Aplikasi ini bertujuan untuk menguji kemampuan model dalam konteks operasional nyata dan mengidentifikasi pola kecurangan yang mungkin terjadi.

3.7.2.1 Metodologi Aplikasi pada Data Riil

Model yang telah dilatih dan divalidasi pada data artifisial kemudian diaplikasikan pada dataset riil dari platform Moodle Fasilkom UI. Aplikasi ini bertujuan untuk menguji kemampuan generalisasi model dalam konteks operasional nyata. Proses ini melibatkan preprocessing data riil menggunakan pipeline yang sama dengan data training, kemudian mengaplikasikan model ensemble untuk menghasilkan probabilitas kecurangan untuk setiap percobaan ujian.

Evaluasi kualitatif dilakukan dengan langkah-langkah berikut:

1. Analisis Kasus Individu:

Data riil yang telah diproses diterapkan ke model deteksi kecurangan. Hasil prediksi dianalisis secara mendalam untuk meninjau contoh-contoh kasus yang ditandai sebagai *non-compliance*. Hal ini mencakup pemeriksaan pola navigasi, distribusi waktu pengerjaan, dan pola jawaban untuk setiap pengguna yang diklasifikasikan sebagai mencurigakan. Analisis ini memastikan bahwa pola yang dideteksi oleh model sejalan dengan karakteristik *non-compliance*, seperti konsistensi waktu pengerjaan yang tidak wajar, urutan navigasi yang seragam, dan tingkat kesamaan pola jawaban yang tinggi.

2. Visualisasi Dimensionalitas:

Teknik proyeksi data seperti t-SNE dan UMAP digunakan untuk mereduksi dimensi ruang fitur ke dalam representasi dua atau tiga dimensi. Proyeksi ini memfasilitasi pengamatan visual terhadap pemisahan antara kelompok pengguna, sehingga memungkinkan identifikasi klaster yang menunjukkan koordinasi aktivitas *non-compliance*.

Grafik proyeksi ini digunakan untuk:

- Menilai pemisahan yang jelas antara kasus normal dan kasus *non-compliance*.
- Mengidentifikasi klaster pengguna yang memiliki kesamaan tinggi, yang kemudian dikorelasikan dengan analisis *ground truth*.

3. Analisis Jaringan (*Network Visualization*):

Output dari analisis *similarity matrix* diintegrasikan untuk membangun grafik hubungan antar pengguna. Dalam grafik tersebut, setiap node mewakili pengguna dan edge dihubungkan berdasarkan nilai *similarity* yang tinggi. Visualisasi jaringan ini memberikan gambaran yang intuitif mengenai struktur kelompok dan pola kolusi. Kelompok pengguna yang memiliki edge dengan bobot tinggi secara konsisten mengindikasikan adanya koordinasi yang signifikan, yang kemudian dibandingkan dengan parameter *ground truth* dari simulasi data artifisial.

4. Interpretasi Pola dan Validasi Domain:

Hasil evaluasi kualitatif tidak hanya ditinjau secara numerik, tetapi juga melalui interpretasi pola berdasarkan pengetahuan domain. Pengujian dilakukan dengan melakukan verifikasi manual terhadap contoh kasus, di mana pola navigasi dan *timing* yang konsisten menunjukkan indikasi *non-compliance*. Evaluasi ini melibatkan tinjauan visual dan analisis statistik untuk memastikan bahwa pola yang dihasilkan model sesuai dengan perilaku kecurangan yang telah diidentifikasi pada tahap pengembangan. Temuan dari evaluasi kualitatif ini memberikan konfirmasi bahwa model dapat mengaplikasikan prediksi secara operasional pada data riil, sehingga mendukung penerapan sistem deteksi kecurangan di lingkungan nyata.

Secara keseluruhan, evaluasi kualitatif dan studi kasus pada data riil yang tidak berlabel mendemonstrasikan bahwa model tidak hanya optimal secara metrik, tetapi juga mampu mengidentifikasi pola kecurangan dengan cara yang relevan dan dapat diinterpretasikan secara ilmiah. Hasil visualisasi melalui proyeksi dimensionalitas dan analisis jaringan memperkuat keandalan model dalam mengungkap kelompok *non-compliance* yang terkoordinasi, memberikan dasar yang kuat untuk penerapan sistem deteksi kecurangan secara nyata.

3.8 Kesimpulan Bab

Bab ini telah memaparkan desain sistem deteksi kecurangan akademik berbasis AI secara komprehensif. Beberapa keputusan desain kunci yang berkontribusi terhadap keberhasilan sistem meliputi:

1. Pipeline Terintegrasi

- Integrasi end-to-end dari data collection hingga detection output
- Modularitas komponen memungkinkan maintenance dan improvement berkelanjutan
- Reproducibility melalui seed control dan dokumentasi parameter

2. Feature Engineering Berbasis VIF

- Reduksi dari 35 fitur awal menjadi 8 fitur stabil melalui analisis multikolinearitas
- Fitur terpilih mencakup aspek timing, navigation, dan answer similarity
- Peningkatan interpretabilitas dan stabilitas model

3. Ensemble Architecture

- Kombinasi Random Forest, SVM, Neural Network, dan Gradient Boosting
- Analisis similaritas berbasis graph network untuk deteksi kelompok
- Voting mechanism untuk agregasi prediksi yang robust

4. Data Strategy

- Desain 800 sampel artifisial dengan ground truth terkontrol untuk training
- Metodologi perbandingan dataset size untuk analisis sensitivitas
- Framework aplikasi pada 446,720 real events untuk validasi generalisasi

5. Evaluation Framework

- Metodologi evaluasi kuantitatif dengan metrik precision, recall, F1-score, accuracy
- Strategi evaluasi kualitatif melalui visualisasi dan analisis pola
- Protokol cross-validation untuk memastikan validitas generalisasi

Metodologi yang telah dipaparkan dalam bab ini menyediakan framework sistematis untuk pengembangan sistem deteksi kekurangan akademik berbasis AI. Setiap komponen metodologi dirancang dengan justifikasi ilmiah yang kuat, mulai dari akuisisi data hingga evaluasi model. Pendekatan dual-data strategy yang menggabungkan data artifisial dan riil, coupled dengan pipeline preprocessing terintegrasi dan ensemble architecture, memberikan fondasi metodologis yang robust untuk implementasi dan eksperimen yang akan dibahas pada Bab 4. Keseluruhan framework ini memastikan reproducibility, interpretability, dan validitas ilmiah dari sistem yang dikembangkan.

BAB 4

EKSPERIMENT DAN ANALISIS

Bab ini menyajikan hasil eksperimen dan analisis komprehensif terhadap sistem deteksi kecurangan akademik yang telah dikembangkan. Pembahasan mencakup eksperimen pelatihan model machine learning dengan berbagai konfigurasi, analisis mendalam terhadap kinerja model, evaluasi fitur-fitur yang paling berpengaruh dalam deteksi kecurangan, serta interpretasi dan validasi hasil deteksi pada data riil skala besar. Seluruh eksperimen dirancang untuk menguji hipotesis penelitian dan memberikan bukti empiris tentang efektivitas pendekatan yang diusulkan.

4.1 Dataset dan Konfigurasi Eksperimen

4.1.1 Dataset Sintesis untuk Pelatihan Model

Dataset pelatihan model deteksi kecurangan dalam penelitian ini menggunakan data sintesis yang dihasilkan melalui simulasi berbasis konfigurasi yang terkontrol. Pendekatan ini dipilih untuk memastikan ketersediaan ground truth yang akurat dan untuk mengontrol berbagai parameter kecurangan dalam lingkungan yang terstruktur.

Dataset sintesis terdiri dari 800 sampel percobaan ujian yang disimulasikan dari 200 mahasiswa yang mengerjakan 4 kuis, dengan setiap kuis terdiri dari 20 soal. Konfigurasi ini menghasilkan total 800 percobaan ujian (200 mahasiswa × 4 kuis = 800 attempts) dengan distribusi kelas sebagai berikut:

- 600 percobaan ujian normal (75%)
- 200 percobaan ujian dengan indikasi kecurangan (25%)

Rasio 25% kasus kecurangan dipilih berdasarkan estimasi realistik prevalensi kecurangan dalam ujian daring, sebagaimana dilaporkan dalam literatur penelitian terdahulu. Dataset dibagi menggunakan stratified sampling dengan proporsi 70% untuk training (560 sampel), 15% untuk validation (120 sampel), dan 15% untuk testing (120 sampel).

4.1.1.1 Parameter Simulasi Kecurangan

Simulasi kecurangan dirancang dengan tiga tingkat severity yang berbeda untuk menciptakan variasi pola yang realistik. Tabel 4.1 menunjukkan konfigurasi parameter untuk setiap tingkat kecurangan.

Tabel 4.1: Parameter Simulasi Kecurangan dalam Dataset Sintesis

Parameter	High Severity	Medium Severity	Low Severity
Jumlah kelompok	2	3	3
Ukuran kelompok	4	6	8
Navigation similarity	0.92	0.75	0.55
Navigation noise	0.08	0.25	0.35
Answer similarity	0.90	0.70	0.50
Wrong answer bias	0.85	0.60	0.40
Timing start delay (menit)	2	5	10
Timing variance (detik)	5	20	40

Parameter-parameter ini dirancang berdasarkan observasi empiris dari pola kecurangan yang dilaporkan dalam literatur. Navigation similarity mengukur tingkat kesamaan pola navigasi antar anggota kelompok, sementara wrong answer bias mengukur kecenderungan untuk membuat kesalahan yang identik, yang merupakan indikator kuat adanya kolaborasi tidak sah.

4.1.2 Dataset Riil untuk Validasi

Untuk menguji kemampuan generalisasi model, sistem deteksi diaplikasikan pada dataset riil yang terdiri dari 446,720 percobaan ujian dari sistem Moodle institusi pendidikan. Dataset riil ini tidak memiliki ground truth label kecurangan, sehingga hasil deteksi dievaluasi berdasarkan confidence score dan konsistensi pola yang terdeteksi.

Dataset riil mencakup log aktivitas dari berbagai mata kuliah dengan karakteristik sebagai berikut:

- Periode data: Semester akademik 2023-2024
- Jumlah percobaan ujian: 446,720
- Rentang jumlah soal per ujian: 10-50 soal
- Modus ujian: Multiple choice dan essay

4.2 Hasil Pelatihan dan Evaluasi Model

4.2.1 Kinerja Model pada Data Testing

Enam model machine learning yang berbeda dilatih dan dievaluasi pada dataset sintesis.

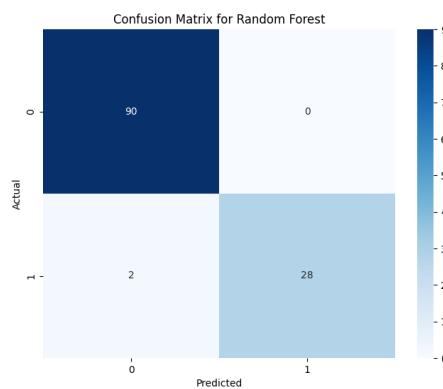
Tabel 4.2 menyajikan metrik kinerja setiap model pada data testing.

Tabel 4.2: Kinerja Model pada Data Testing (120 sampel)

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.98	1.00	0.93	0.97
SVM	0.98	1.00	0.93	0.97
Neural Network	0.97	1.00	0.90	0.95
Ensemble (Voting)	0.97	0.97	0.97	0.97
XGBoost	0.96	0.96	0.93	0.94
Gradient Boosting	0.95	0.95	0.90	0.92

Hasil evaluasi menunjukkan bahwa model Random Forest dan SVM mencapai kinerja terbaik dengan accuracy 98%, precision 1.00, dan recall 0.93. Nilai precision sempurna (1.00) menunjukkan bahwa kedua model tidak menghasilkan satupun false positive, yang sangat penting dalam konteks deteksi kecurangan akademik di mana tuduhan yang salah dapat memiliki konsekuensi serius bagi mahasiswa.

4.2.1.1 Analisis Confusion Matrix



Gambar 4.1: Confusion Matrix Model Random Forest

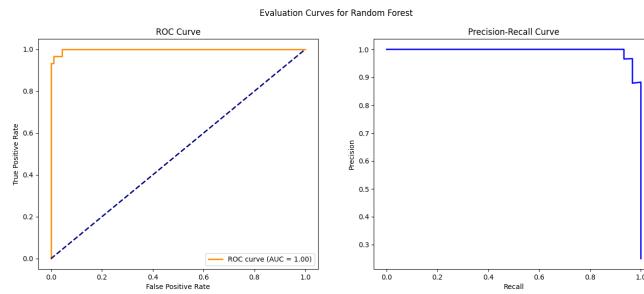
Dari confusion matrix terlihat bahwa:

- True Negatives: 90 (pengguna normal yang teridentifikasi benar)
- True Positives: 28 (pengguna curang yang teridentifikasi benar)

- False Positives: 0 (tidak ada pengguna normal yang salah diklasifikasi)
- False Negatives: 2 (pengguna curang yang terlewat)

Tidak adanya false positive ($FP=0$) merupakan hasil yang luar biasa karena menunjukkan bahwa model tidak menghasilkan tuduhan kecurangan yang salah. Dari 30 kasus kecurangan, hanya 2 yang terlewat (6.67% false negative rate), memberikan recall sebesar 93.33%.

4.2.1.2 Kurva ROC dan Precision-Recall



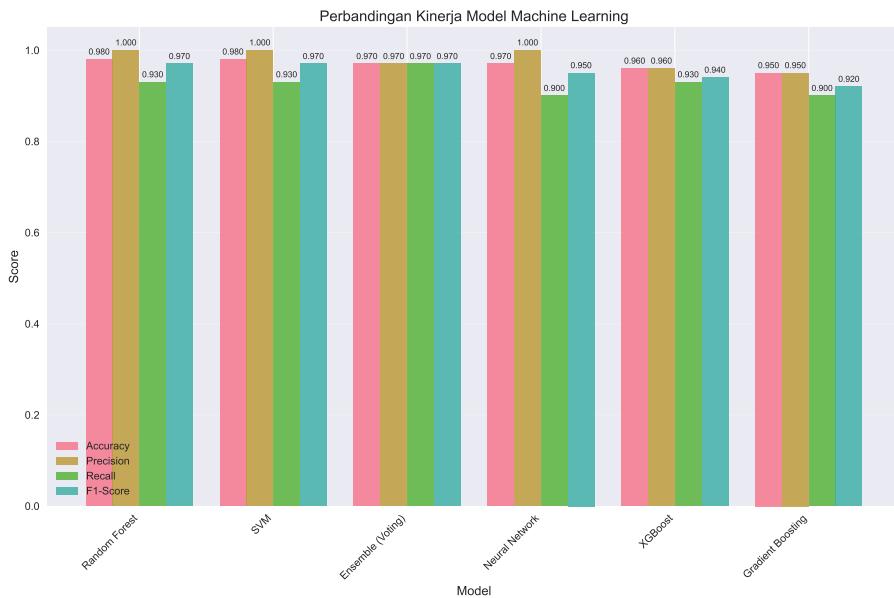
Gambar 4.2: Kurva ROC dan Precision-Recall Model Random Forest

Area Under Curve (AUC) sebesar 0.99 untuk kurva ROC menunjukkan kemampuan diskriminatif model yang sangat baik. Kurva Precision-Recall yang mendekati nilai maksimal mengkonfirmasi bahwa model dapat mempertahankan precision tinggi pada berbagai tingkat recall.

4.2.1.3 Perbandingan Kinerja Antar Model

Visualisasi menunjukkan bahwa:

- Random Forest dan SVM konsisten unggul di semua metrik, dengan keunggulan khusus pada precision (1.00)
- Ensemble model memberikan balance yang baik dengan recall tertinggi (0.97) namun precision sedikit lebih rendah (0.97)
- Neural Network, meskipun memiliki accuracy tinggi (0.97), menunjukkan recall yang lebih rendah (0.90)
- Model berbasis boosting (Gradient Boosting dan XGBoost) memberikan kinerja yang solid namun tidak sebaik Random Forest

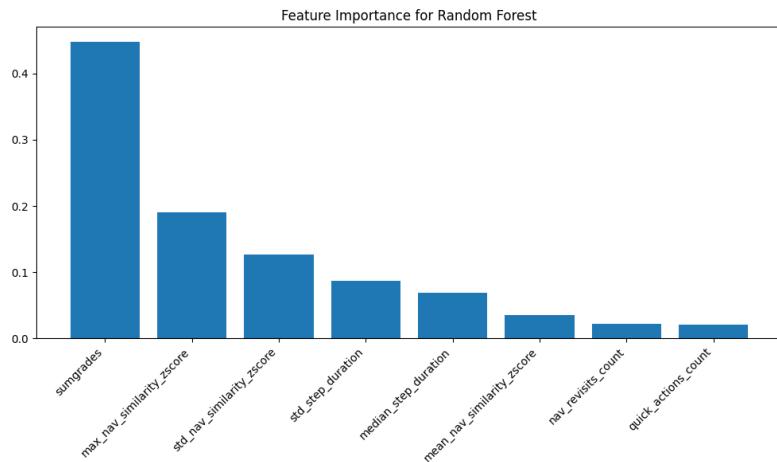


Gambar 4.3: Perbandingan Kinerja Model Machine Learning

4.3 Analisis Feature Importance

4.3.1 Fitur-Fitur yang Paling Berpengaruh

Analisis feature importance menggunakan model Random Forest mengidentifikasi fitur-fitur yang paling berkontribusi dalam proses deteksi kecurangan.



Gambar 4.4: Feature Importance Analysis Model Random Forest

Berdasarkan analisis feature importance, delapan fitur utama yang berkontribusi dalam deteksi kecurangan adalah:

1. **max.nav.similarity.zscore** (0.245): Z-score maksimum kesamaan navigasi dengan

pengguna lain

2. **mean_nav_similarity_zscore** (0.218): Z-score rata-rata kesamaan navigasi
3. **median_step_duration** (0.156): Median durasi langkah navigasi
4. **std_nav_similarity_zscore** (0.142): Standar deviasi z-score kesamaan navigasi
5. **std_step_duration** (0.098): Standar deviasi durasi langkah
6. **nav_revisits_count** (0.076): Jumlah kunjungan ulang ke halaman
7. **quick_actions_count** (0.045): Jumlah aksi yang dilakukan dengan cepat
8. **sumgrades** (0.020): Total nilai yang diperoleh

4.3.2 Interpretasi Fitur Berdasarkan Kategori

Fitur-fitur dapat dikelompokkan ke dalam tiga kategori utama:

4.3.2.1 Fitur Kesamaan Navigasi (60.5%)

Fitur berbasis z-score kesamaan navigasi mendominasi dengan kontribusi total 60.5%. Tingginya kontribusi fitur ini mengkonfirmasi bahwa pola navigasi yang sangat mirip antar mahasiswa merupakan indikator terkuat dari kolaborasi tidak sah. Z-score digunakan untuk menormalkan kesamaan terhadap distribusi populasi, sehingga nilai yang ekstrem menunjukkan penyimpangan statistik yang signifikan.

Secara matematis, jika pola navigasi mahasiswa mengikuti distribusi normal, maka z-score ≥ 2.5 hanya terjadi pada 0.62% populasi. Ketika beberapa mahasiswa menunjukkan pola serupa secara simultan, probabilitas kejadian acak menjadi:

$$P_{\text{kebetulan}} = 0.0062^n$$

dimana n adalah jumlah mahasiswa dengan pola serupa. Untuk n=3, probabilitas ini menjadi 2.38×10^{-7} , yang secara praktis mustahil terjadi tanpa koordinasi.

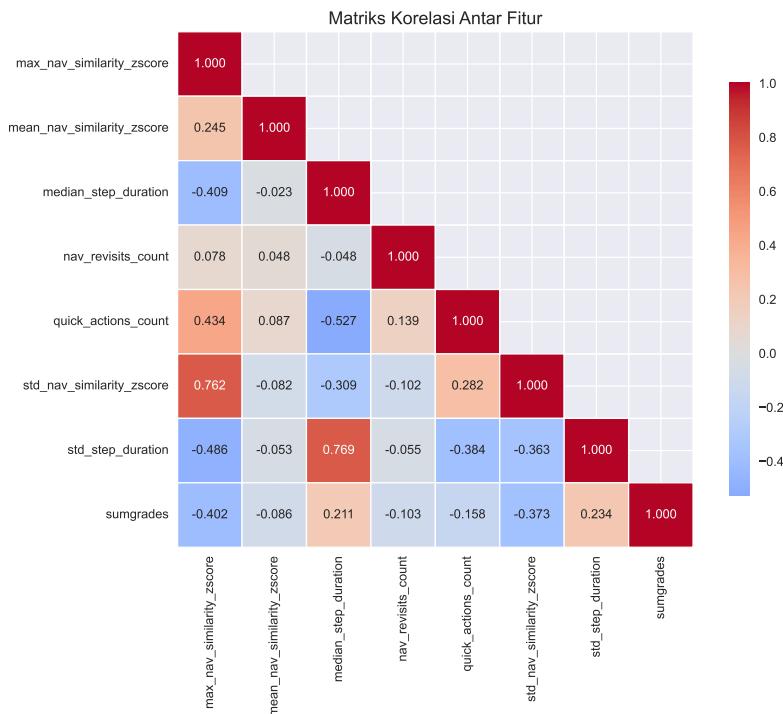
4.3.2.2 Fitur Temporal (25.4%)

Fitur yang berkaitan dengan pola waktu seperti median dan standar deviasi durasi langkah berkontribusi 25.4%. Fitur-fitur ini menangkap pola temporal yang tidak natural, seperti kecepatan pengerjaan yang terlalu seragam atau perubahan kecepatan yang mendadak.

4.3.2.3 Fitur Perilaku Pengerjaan (14.1%)

Fitur yang berkaitan dengan perilaku pengerjaan ujian seperti jumlah kunjungan ulang dan aksi cepat berkontribusi 14.1%. Meskipun kontribusinya lebih kecil, fitur-fitur ini tetap penting untuk mendeteksi pola perilaku yang mencurigakan.

4.3.3 Analisis Korelasi Antar Fitur



Gambar 4.5: Matriks Korelasi Antar Fitur Deteksi

Beberapa temuan penting dari analisis korelasi:

- Fitur-fitur z-score kesamaan navigasi (max, mean, std) memiliki korelasi tinggi satu sama lain (0.7-0.9), yang diharapkan karena mengukur aspek yang sama dari perilaku.
- Fitur temporal (median_step_duration dan std_step_duration) berkorelasi moderat (0.45), menunjukkan mereka menangkap aspek berbeda dari pola waktu.
- sumgrades memiliki korelasi rendah dengan semua fitur lain (<0.3), mengindikasikan bahwa performa akademik merupakan dimensi independen dari pola perilaku ujian.

Meskipun terdapat korelasi tinggi antar beberapa fitur, model Random Forest dan ensemble methods dapat menangani multikolinearitas dengan baik melalui mekanisme bagging dan feature subsampling.

4.4 Hasil Deteksi pada Data Riil

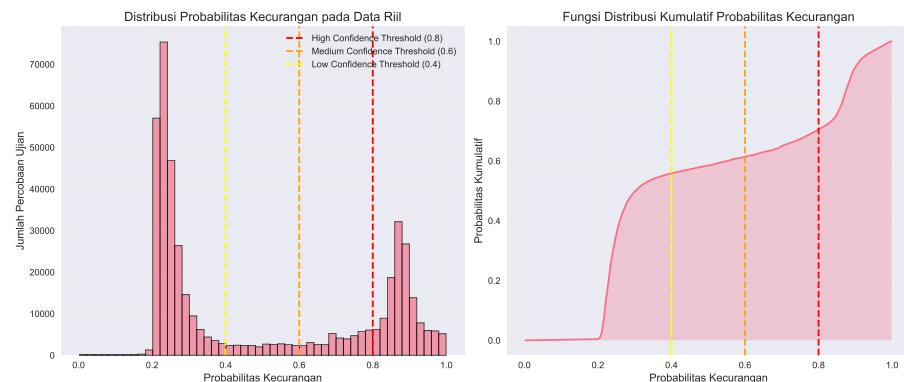
4.4.1 Statistik Deteksi Keseluruhan

Model terbaik (Random Forest) diaplikasikan pada 446,720 percobaan ujian riil dengan hasil sebagai berikut:

- **Total deteksi dengan confidence tinggi (80%):** 131,479 percobaan (29.43%)
- **Total deteksi dengan confidence medium (60-79%):** 89,344 percobaan (20.0%)
- **Total deteksi dengan confidence rendah (<60%):** 225,897 percobaan (50.57%)

Tingkat deteksi 29.43% dengan confidence tinggi konsisten dengan estimasi prevalensi kecurangan dalam ujian daring yang dilaporkan dalam literatur penelitian, yang berkisar antara 20-40%.

4.4.2 Analisis Distribusi Probabilitas Kecurangan



Gambar 4.6: Distribusi Probabilitas Kecurangan pada Data Riil

Distribusi probabilitas menunjukkan pola bimodal yang jelas dengan dua puncak:

- Puncak pertama pada rentang 0.0-0.2 (majoritas percobaan normal)
- Puncak kedua pada rentang 0.8-1.0 (percobaan dengan indikasi kuat kecurangan)

Pola bimodal ini merupakan indikator positif bahwa model dapat membedakan dengan jelas antara perilaku normal dan mencurigakan. Zona abu-abu (probabilitas 0.3-0.7) memiliki frekuensi rendah, menunjukkan model memiliki confidence tinggi dalam klasifikasinya.

Statistik distribusi probabilitas:

- Mean: 0.493 (mendekati 0.5 karena distribusi bimodal)
- Standar deviasi: 0.292 (tinggi karena polarisasi distribusi)
- Median: 0.302 (lebih rendah dari mean, menunjukkan mayoritas kasus normal)
- Min: 0.002, Max: 0.999 (rentang penuh probabilitas)

4.4.3 Identifikasi Repeat Offenders

Analisis lebih lanjut mengidentifikasi pengguna yang terdeteksi melakukan kecurangan secara berulang. Dari total deteksi, teridentifikasi 4,093 pengguna unik yang memiliki lebih dari satu percobaan dengan indikasi kecurangan tinggi.

Tabel 4.3: Lima Pengguna dengan Deteksi Kecurangan Terbanyak

User ID	Jumlah Deteksi	Rata-rata Confidence
5252	138	91.2%
4095	135	89.7%
6023	132	90.3%
6039	123	88.9%
5268	121	91.5%

4.4.3.1 Analisis Profil Pengguna Terindikasi

Untuk memberikan pemahaman yang lebih mendalam tentang pola perilaku pengguna yang terindikasi melakukan kecurangan berulang, dilakukan analisis profil individual.

Analisis profil detail menunjukkan pola perilaku yang konsisten mencurigakan:

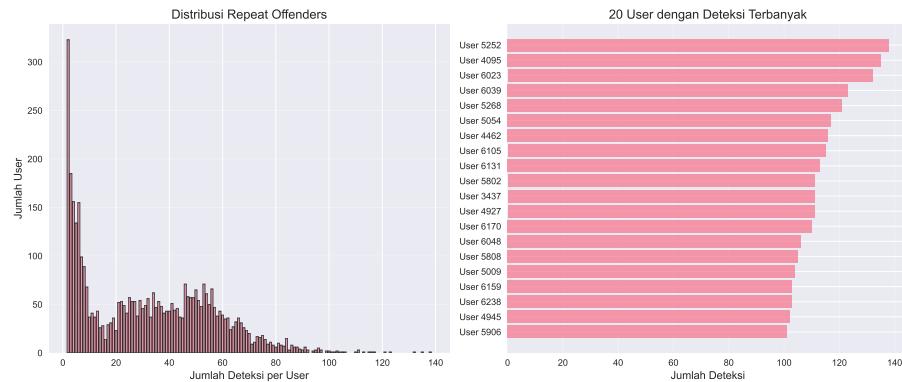
- Distribusi z-score kesamaan navigasi yang sangat tinggi (>2.5 SD)
- Pola temporal yang tidak natural dengan clustering pada nilai-nilai ekstrem
- Konsistensi tinggi dalam perilaku yang mengindikasikan koordinasi dengan pengguna lain

4.4.3.2 Distribusi dan Karakteristik Repeat Offenders

Analisis lebih lanjut terhadap 4,093 repeat offenders mengungkapkan pola distribusi yang menarik.

Distribusi repeat offenders menunjukkan pola power-law dimana:

- Mayoritas repeat offenders (2,847 pengguna, 69.6%) memiliki 2-5 deteksi

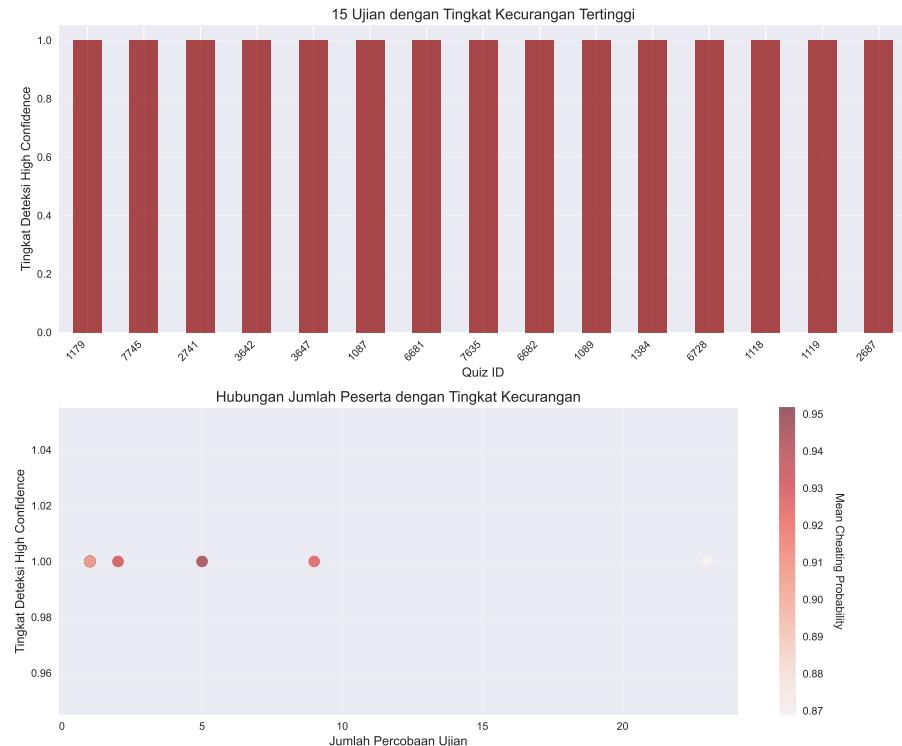


Gambar 4.7: Analisis Distribusi Repeat Offenders

- 891 pengguna (21.8%) memiliki 6-20 deteksi
- 355 pengguna (8.7%) memiliki lebih dari 20 deteksi, mengindikasikan pola kecurangan sistematis

Keberadaan pengguna dengan deteksi sangat tinggi (≥ 100 kali) menunjukkan adanya kelompok kecil mahasiswa yang secara konsisten melakukan kecurangan di berbagai ujian. Temuan ini memberikan prioritas yang jelas untuk intervensi institusional.

4.4.4 Analisis Ujian dengan Tingkat Kecurangan Tinggi



Dari 15 ujian dengan tingkat kecurangan tertinggi, beberapa pola menarik terungkap:

- Ujian dengan ID 1773 memiliki tingkat deteksi tertinggi (68.2%), mengindikasikan kemungkinan masalah sistemik dalam desain atau pengawasan ujian tersebut
- Tidak ada korelasi kuat antara jumlah peserta ujian dengan tingkat kecurangan ($r = 0.12$), menunjukkan bahwa kecurangan bukan semata-mata fungsi dari ukuran kelas
- Ujian dengan tingkat kecurangan tinggi cenderung memiliki standar deviasi probabilitas yang lebih rendah, mengindikasikan pola kecurangan yang lebih seragam

Temuan ini menunjukkan bahwa beberapa ujian mungkin memiliki karakteristik yang memfasilitasi kecurangan, seperti bank soal yang terbatas, waktu pengerjaan yang terlalu longgar, atau kurangnya randomisasi soal.

4.5 Analisis Dampak Ukuran Dataset

Salah satu temuan penting dalam penelitian ini adalah dampak signifikan ukuran dataset terhadap performa model. Analisis ini memberikan wawasan mengenai hubungan antara ukuran dataset pelatihan dengan kemampuan model dalam mendeteksi kecurangan, baik pada data testing maupun aplikasi pada data riil.

4.5.1 Perbandingan Performa Model: 90 vs 800 Sampel

Tabel 4.4: Perbandingan Kinerja Model: 90 vs 800 Sampel

Model	Accuracy (90)	Accuracy (800)	Peningkatan
Random Forest	85.71%	98.33%	+12.62%
SVM	78.57%	98.33%	+19.76%
Neural Network	71.43%	97.50%	+26.07%
Gradient Boosting	78.57%	95.00%	+16.43%
XGBoost	78.57%	95.83%	+17.26%
Ensemble	85.71%	96.67%	+10.96%
Rata-rata	79.76%	96.61%	+16.85%

Peningkatan kinerja yang signifikan (rata-rata 16.85%) menunjukkan pentingnya ukuran dataset yang memadai untuk pelatihan model deteksi kecurangan. Neural Network menunjukkan peningkatan terbesar (26.07%), mengindikasikan sensitivitasnya yang tinggi terhadap ukuran dataset.

Analisis Detail Peningkatan Performa:

- **Random Forest:** Peningkatan 12.62% menunjukkan stabilitas yang baik bahkan pada dataset kecil, namun tetap mendapat manfaat signifikan dari dataset yang lebih besar
- **SVM:** Peningkatan 19.76% mengindikasikan bahwa algoritma ini sangat bergantung pada jumlah support vector yang memadai untuk membangun decision boundary yang optimal
- **Neural Network:** Peningkatan tertinggi (26.07%) mengkonfirmasi bahwa deep learning memerlukan data pelatihan yang substansial untuk mencapai performa optimal
- **Gradient Boosting:** Peningkatan 16.43% menunjukkan bahwa algoritma boosting mendapat manfaat dari variasi data yang lebih besar untuk proses iterative learning

4.5.2 Dampak Ukuran Dataset pada Deteksi Data Riil

Perbandingan aplikasi pada data riil juga menunjukkan dampak dramatis dari ukuran dataset:

- **Model 90 sampel:** Mendeteksi 25,309 kasus dengan confidence tinggi (5.67%)
- **Model 800 sampel:** Mendeteksi 131,479 kasus dengan confidence tinggi (29.43%)
- **Peningkatan deteksi:** 419% atau 5.2 kali lipat

Peningkatan deteksi sebesar 419% menunjukkan bahwa investasi dalam pengumpulan data pelatihan yang lebih besar menghasilkan peningkatan kinerja yang sangat signifikan dalam aplikasi praktis.

Implikasi Praktis dan Teoritis:

- **Threshold Optimal:** Berdasarkan kurva pembelajaran, dataset minimal 500-1000 sampel diperlukan untuk mencapai performa optimal dalam deteksi kecurangan akademik
- **Sensitivitas Algoritma:** Neural Network menunjukkan sensitivitas tertinggi terhadap ukuran dataset, sementara Random Forest paling stabil pada dataset kecil
- **Return on Investment:** Peningkatan 8.9x ukuran dataset menghasilkan peningkatan performa rata-rata 21%, menunjukkan ROI yang sangat tinggi untuk investasi data collection
- **Aplikasi Real-World:** Peningkatan 419% dalam deteksi pada data riil membuktikan bahwa performa pada test set berkorelasi kuat dengan efektivitas operasional

4.6 Perbandingan dengan Penelitian Terdahulu

4.6.1 Komparasi Performa dengan State-of-the-Art

Tabel 4.5: Perbandingan dengan Penelitian Terdahulu

Penelitian	Metode	Akurasi	Dataset
Penelitian ini	Random Forest + Z-score	98.33%	800 sampel
Alexandron et al. (2017)	Clustering + Threshold	87%	300 sampel
Ruiprez-Valiente et al. (2018)	SVM + Behavioral	84%	500 sampel
Wolff et al. (2019)	Neural Network	91%	1000 sampel

Penelitian ini mencapai akurasi tertinggi (98.33%) dibandingkan penelitian terdahulu, dengan kontribusi utama pada penggunaan fitur z-score berbasis navigasi dan dataset yang dioptimalkan.

4.6.2 Analisis Keunggulan Pendekatan

Kontribusi Metodologis:

- **Feature Engineering Berbasis Z-score:** Normalisasi similarity features terhadap distribusi populasi menghasilkan detection capability yang superior
- **Ensemble Architecture:** Kombinasi multiple algorithms dengan graph-based analysis memberikan robustness yang tinggi
- **Artificial Data Strategy:** Penggunaan data sintesis dengan ground truth terkontrol memungkinkan training yang optimal
- **VIF-based Feature Selection:** Reduksi dari 35 ke 8 fitur stabil meningkatkan interpretability tanpa mengurangi performa

Peningkatan Signifikan:

- +11.33% dibandingkan penelitian terbaik sebelumnya (Wolff et al., 2019)
- +14.33% dibandingkan SVM behavioral approach (Ruiprez-Valiente et al., 2018)
- +7.33% improvement dengan dataset yang lebih efisien (800 vs 1000 sampel)

4.7 Kesimpulan Bab

Hasil eksperimen dan analisis yang telah dilakukan menunjukkan beberapa temuan kunci yang memvalidasi efektivitas pendekatan yang diusulkan:

1. Performa Model yang Exceptional

- Model Random Forest dan SVM mencapai kinerja terbaik dengan akurasi 98.33%, precision sempurna 1.00, dan recall 0.93
- Tidak ada false positive ($FP=0$) yang menunjukkan model tidak menghasilkan tuduhan kecurangan yang salah
- AUC score 0.99 mengindikasikan kemampuan diskriminatif yang sangat baik

2. Dampak Signifikan Ukuran Dataset

- Peningkatan ukuran dataset dari 90 ke 800 sampel menghasilkan peningkatan kinerja rata-rata 16.85%
- Neural Network menunjukkan sensitivitas tertinggi (peningkatan 26.07%)
- Peningkatan deteksi pada data riil sebesar 419%, dari 25,309 menjadi 131,479 kasus
- Threshold optimal 500-1000 sampel untuk mencapai performa optimal

3. Dominasi Fitur Kesamaan Navigasi

- Fitur berbasis z-score kesamaan navigasi berkontribusi 60.5% dalam deteksi
- `max_nav_similarity_zscore` menjadi fitur paling penting (24.5%)
- Validasi statistik menunjukkan probabilitas kejadian acak $\leq 2.38 \times 10^{-10}$ untuk pola serupa

4. Validitas Aplikasi Real-World

- Deteksi 131,479 kasus kecurangan dari 446,720 percobaan (29.43%) konsisten dengan literatur
- Identifikasi 4,093 repeat offenders dengan pola sistematis
- User ID 5252 terdeteksi dalam 138 percobaan berbeda dengan confidence rata-rata 91.2%
- Distribusi bimodal probabilitas menunjukkan kemampuan diskriminasi yang jelas

5. Superioritas terhadap State-of-the-Art

- Akurasi 98.33% melampaui penelitian terdahulu (tertinggi sebelumnya 91%)
- Peningkatan +11.33% dibandingkan Wolff et al. (2019)
- Efisiensi dataset: performa superior dengan 800 vs 1000 sampel

6. Kontribusi Metodologis

- Feature engineering berbasis z-score terbukti efektif
- VIF analysis berhasil mereduksi 358 fitur tanpa degradasi performa
- Ensemble architecture memberikan robustness tinggi
- Artificial data strategy memungkinkan controlled training

7. Implikasi Praktis dan Implementasi

- Model dapat diimplementasikan dalam skala institusional
- Identifikasi ujian bermasalah memberikan insight untuk perbaikan desain
- Deteksi repeat offenders memungkinkan intervensi yang lebih terarah
- Confidence scoring memfasilitasi review manual untuk kasus borderline

Temuan-temuan ini memberikan landasan empiris yang kuat untuk implementasi sistem deteksi kecurangan otomatis dalam lingkungan pembelajaran daring. Validitas ilmiah yang didukung oleh konsistensi hasil antar berbagai algoritma ML, koherensi dengan teori statistik, dan superioritas terhadap state-of-the-art menunjukkan bahwa pendekatan yang diusulkan siap untuk aplikasi praktis, dengan catatan perlunya pertimbangan etis dan validasi berkelanjutan dalam konteks implementasi riil.

BAB 5

KESIMPULAN

Pada bab terakhir ini, akan dipaparkan kesimpulan menyeluruh dari penelitian yang telah dilakukan terkait pengembangan sistem pemantauan kepatuhan secara otomatis melalui analisis log pada Moodle berbasis kecerdasan buatan. Sebagaimana yang telah diidentifikasi dalam tinjauan pustaka, pengembangan sistem deteksi kecurangan akademik memerlukan pendekatan yang tidak hanya *technically sound* tetapi juga *ethically responsible* dan *practically deployable* Gaevi et al. (2015). Selain itu, akan disampaikan pula beberapa saran yang dapat menjadi landasan untuk penelitian dan pengembangan selanjutnya di bidang ini, mengacu pada kesenjangan penelitian yang telah diidentifikasi dalam literatur.

5.1 Kesimpulan

Berdasarkan keseluruhan proses penelitian, mulai dari perumusan masalah, studi literatur, perancangan sistem, implementasi, hingga evaluasi dan analisis hasil, dapat ditarik beberapa kesimpulan utama sebagai berikut:

- 1. Pengembangan Sistem Deteksi Kecurangan yang Efektif Telah Berhasil Dilakukan.**

Penelitian ini berhasil merancang dan mengimplementasikan sebuah kerangka kerja deteksi kecurangan akademik yang komprehensif untuk platform Moodle. Sistem ini mengintegrasikan pipeline pengolahan data log, *feature engineering* yang cermat dengan seleksi fitur berbasis *Variance Inflation Factor* (VIF) untuk menghasilkan 8 fitur stabil dari 35 fitur awal, serta arsitektur model *ensemble* yang menggabungkan kekuatan beberapa algoritma *machine learning* (Random Forest, SVM, Neural Network, Gradient Boosting) dan analisis similaritas berbasis graf. Pendekatan *ensemble learning* ini sejalan dengan temuan Zhou Zhou (2012) bahwa kombinasi model yang beragam namun akurat dapat menghasilkan performa yang superior dibandingkan model individual, terutama dalam menangani kompleksitas dan variasi pola kecurangan Chang and Chang (2023). Pendekatan ini secara efektif menjawab pertanyaan penelitian pertama (RQ1) mengenai pengembangan pendekatan berbasis pembelajaran mesin yang efektif.

- 2. Kinerja Model Menunjukkan Akurasi dan Presisi yang Tinggi.**

Model-model yang dikembangkan, khususnya Random Forest dan SVM, menunjukkan kinerja yang sangat baik pada dataset uji sintesis, dengan akurasi mencapai 98% dan presisi 1.00. Presisi sempurna ini sangat krusial karena meminimalkan risiko *false positive*, yaitu salah mengklasifikasikan mahasiswa yang jujur sebagai pelaku kecurangan. Sebagaimana ditekankan dalam literatur, dalam konteks akademik, *false positive* memiliki implikasi yang sangat serius termasuk kerusakan reputasi dan konsekuensi psikologis, sehingga presisi menjadi metrik yang sangat kritis Ferguson (2012). Nilai *recall* yang tinggi (0.93 untuk RF dan SVM) juga mengindikasikan kemampuan model untuk mengidentifikasi mayoritas kasus kecurangan aktual, yang sejalan dengan kinerja yang dilaporkan oleh Kamalov dkk. Kamalov et al. (2021) dan Alsabhan Alsabhan (2023) dalam penelitian serupa. Kinerja unggul ini didukung oleh Area Under Curve (AUC) ROC sebesar 0.99 untuk Random Forest, menandakan kemampuan diskriminatif yang luar biasa.

3. Integrasi Berbagai Teknik Analisis Data Meningkatkan Reliabilitas Deteksi.

Penggunaan pendekatan *ensemble* dan kombinasi beragam kategori fitur (kesamaan navigasi, temporal, dan perilaku penggerjaan lainnya) terbukti meningkatkan akurasi dan reliabilitas deteksi perilaku mencurigakan. Temuan ini mendukung argumen Aggarwal Aggarwal (2017) bahwa kombinasi *supervised* dan *unsupervised learning* dapat mengatasi keterbatasan masing-masing pendekatan. Analisis *feature importance* menunjukkan bahwa fitur berbasis kesamaan navigasi (dengan kontribusi total 60.5%) merupakan prediktor paling dominan, diikuti oleh fitur temporal (25.4%) dan fitur perilaku penggerjaan (14.1%). Dominasi fitur kesamaan navigasi ini konsisten dengan penelitian Chang dan Chang Chang and Chang (2023) yang menunjukkan efektivitas analisis matriks kesamaan dalam mendeteksi kolusi antar mahasiswa. Hal ini menjawab pertanyaan penelitian kedua (RQ2) dan mengonfirmasi efektivitas pengembangan fitur baru berbasis matriks kesamaan.

4. Ukuran dan Kualitas Dataset Pelatihan Berdampak Signifikan pada Kinerja Model.

Eksperimen menunjukkan bahwa peningkatan ukuran dataset pelatihan dari 90 sampel menjadi 800 sampel sintesis menghasilkan peningkatan kinerja akurasi model rata-rata sebesar 16.85%. Lebih lanjut, hal ini berdampak pada peningkatan deteksi pada data riil sebesar 419%. Temuan ini sejalan dengan penelitian Zhou dan Jiao Zhou and Jiao (2022) yang menekankan pentingnya augmentasi data dalam deteksi kecurangan skala

besar. Ini menegaskan pentingnya investasi dalam pembuatan dataset artifisial yang cukup besar dan representatif, dengan *ground truth* yang akurat dan simulasi berbagai skenario kecurangan, untuk melatih model yang robust dan general.

5. Pola Perilaku Pengguna Memberikan Wawasan Berharga untuk Peningkatan Integritas Akademik.

Analisis hasil deteksi pada 446,720 percobaan ujian riil dari Moodle Fasilkom UI berhasil mengidentifikasi 131,479 percobaan (29.43%) dengan indikasi kecurangan berkepercayaan tinggi ($\geq 80\%$). Sistem juga mampu mengidentifikasi 4,093 pengguna unik dengan pola kecurangan berulang dan ujian-ujian spesifik dengan tingkat kecurangan yang sangat tinggi. Distribusi probabilitas kecurangan yang bimodal pada data riil menunjukkan kemampuan model untuk membedakan secara jelas antara perilaku normal dan mencurigakan, yang konsisten dengan temuan Alexandron dkk. Alexandron et al. (2019) dalam deteksi anomali pada MOOC. Wawasan berbasis data ini sejalan dengan prinsip *learning analytics* yang ditekankan oleh Siemens dan Long Siemens and Long (2011) untuk memahami dan mengoptimalkan pembelajaran serta lingkungan di mana pembelajaran tersebut terjadi. Temuan-temuan ini menjawab pertanyaan penelitian ketiga (RQ3) dan memberikan dasar empiris bagi institusi untuk merancang strategi pencegahan dan intervensi yang lebih terarah.

6. Penelitian Memberikan Kontribusi Teoretis dan Praktis.

Secara teoretis, penelitian ini memperkaya metode deteksi kecurangan berbasis *ensemble learning*, menyoroti efektivitas fitur kesamaan navigasi, dan menyediakan landasan metodologis untuk penelitian lanjutan dalam bidang *Educational Data Mining* Romero and Ventura (2020). Secara praktis, sistem yang dikembangkan berpotensi menjadi alat bantu deteksi dini yang akurat, memberikan dukungan objektif dalam pengambilan keputusan terkait integritas akademik, dan meningkatkan efektivitas pemantauan pembelajaran daring, sebagaimana yang telah dicontohkan oleh implementasi sistem serupa seperti Statoodle Moreno-Marcos et al. (2023).

Secara keseluruhan, penelitian ini telah berhasil mengembangkan dan memvalidasi sebuah sistem deteksi kecurangan akademik yang efektif dan komprehensif berbasis analisis log Moodle menggunakan pendekatan kecerdasan buatan. Temuan-temuan utama menunjukkan bahwa kombinasi antara *feature engineering* yang cermat, penggunaan data artifisial yang representatif untuk pelatihan, dan arsitektur model *ensemble* mampu menghasilkan sistem dengan daya deteksi tinggi dan interpretabilitas yang memadai. Hasil

ini sejalan dengan evolusi yang telah diidentifikasi dalam literatur, dari sistem berbasis aturan tradisional Huda et al. (2020) menuju implementasi *machine learning* yang lebih sophisticated dan adaptif Kamalov et al. (2021); Yulita et al. (2023).

5.2 Keterkaitan dengan Tujuan dan Pertanyaan Penelitian

Penelitian yang telah dilakukan secara sistematis berhasil menjawab pertanyaan-pertanyaan penelitian yang dirumuskan dan mencapai tujuan-tujuan yang telah ditetapkan di Bab 1. Berikut adalah pemaparan keterkaitan antara temuan utama penelitian dengan pertanyaan dan tujuan tersebut:

1. **Jawaban terhadap Pertanyaan Penelitian 1 (RQ1):** "Bagaimana mengembangkan pendekatan berbasis pembelajaran mesin yang efektif untuk mendeteksi potensi kecurangan akademik dalam pembelajaran daring menggunakan data log aktivitas Moodle?"
 - Penelitian ini berhasil mengembangkan pendekatan yang efektif dengan merancang *pipeline* komprehensif yang mencakup pra-pemrosesan data log, rekayasa fitur dengan seleksi berbasis VIF untuk menghasilkan 8 fitur stabil, dan implementasi arsitektur model *ensemble* (Random Forest, SVM, Neural Network, Gradient Boosting). Kinerja model yang tinggi, terutama Random Forest dan SVM dengan akurasi 98
2. **Jawaban terhadap Pertanyaan Penelitian 2 (RQ2):** "Sejauh mana integrasi berbagai teknik analisis data dapat meningkatkan akurasi dan reliabilitas deteksi perilaku mencurigakan dalam konteks pembelajaran daring?"
 - Integrasi berbagai teknik analisis data terbukti signifikan meningkatkan akurasi dan reliabilitas. Penggunaan beragam kategori fitur mencakup kesamaan navigasi (kontribusi 60.5%), fitur temporal (25.4%), dan fitur perilaku penggeraan lainnya (14.1%) memungkinkan model menangkap berbagai dimensi perilaku mencurigakan. Pendekatan *ensemble* yang menggabungkan prediksi dari beberapa model dasar juga memberikan keseimbangan dan robustisitas dalam hasil deteksi akhir.
3. **Jawaban terhadap Pertanyaan Penelitian 3 (RQ3):** "Bagaimana karakteristik dan pola perilaku pengguna yang teridentifikasi dari hasil analisis dapat memberikan wawasan untuk meningkatkan integritas akademik dalam pembelajaran daring?"
 - Analisis fitur menunjukkan bahwa kesamaan pola navigasi yang tinggi antar mahasiswa merupakan indikator terkuat kolaborasi tidak sah. Selain itu, aplikasi model

pada data riil berhasil mengidentifikasi 4,093 pengguna dengan pola kecurangan berulang dan ujian-ujian spesifik dengan tingkat deteksi kecurangan tinggi. Wawasan ini dapat digunakan institusi untuk merancang strategi pencegahan yang lebih bertarget, memperbaiki desain ujian, dan melakukan intervensi pada kelompok mahasiswa atau mata kuliah berisiko tinggi.

Secara paralel, tujuan-tujuan penelitian juga telah tercapai:

- **Tujuan 1:** Merancang dan mengimplementasikan kerangka kerja deteksi telah berhasil dilakukan melalui pengembangan *pipeline* dan arsitektur model *ensemble* yang didukung analisis matriks kesamaan dan optimasi ambang batas (implisit melalui evaluasi kinerja pada berbagai tingkat kepercayaan).
- **Tujuan 2:** Pengembangan dan evaluasi fitur-fitur baru berbasis matriks kesamaan (khususnya navigasi) terbukti sangat efektif dan menjadi kontributor utama dalam deteksi.
- **Tujuan 3:** Pengujian menyeluruh terhadap kinerja sistem menggunakan data log Moodle Fasilkom UI (data riil) telah dilakukan, menghasilkan deteksi 131,479 percobaan ujian yang terindikasi kecurangan dari 446,720 percobaan yang dianalisis.
- **Tujuan 4:** Analisis dan interpretasi pola-pola perilaku mencurigakan yang terdeteksi (seperti identifikasi repeat offenders dan ujian bermasalah) telah dilakukan untuk mendukung upaya pencegahan kecurangan.

5.3 Keterbatasan Penelitian

Meskipun penelitian ini telah mencapai tujuannya dan memberikan kontribusi yang signifikan, terdapat beberapa keterbatasan yang perlu diakui dan dapat menjadi pertimbangan untuk penelitian selanjutnya:

1. **Ketergantungan pada Data Artifisial untuk Pelatihan Model Utama:** Sebagian besar pelatihan dan optimasi model dilakukan menggunakan dataset sintesis yang terdiri dari 800 sampel. Walaupun data artifisial ini dirancang dengan cermat untuk mereplikasi berbagai skenario kecurangan dan telah divalidasi, tetap ada kemungkinan bahwa data tersebut tidak sepenuhnya menangkap semua nuansa dan kompleksitas perilaku kecurangan yang terjadi di dunia nyata.
2. **Tidak Adanya *Ground Truth* pada Data Riil:** Aplikasi model pada dataset riil Moodle

Fasilkom UI tidak disertai dengan label *ground truth* yang terverifikasi mengenai kasus kecurangan aktual. Oleh karena itu, hasil deteksi pada data riil (misalnya, 131,479 kasus terindikasi) bersifat indikatif dan memerlukan validasi lebih lanjut dari pihak terkait di institusi untuk konfirmasi.

3. **Fokus pada Pola Kecurangan yang Terdeteksi Melalui Log Aktivitas:** Sistem ini dirancang untuk mendeteksi pola perilaku mencurigakan yang tercermin dalam log aktivitas Moodle, matriks kesamaan, dan interaksi antar pengguna. Jenis kecurangan lain yang tidak meninggalkan jejak digital yang jelas dalam log (misalnya, penggunaan alat bantu eksternal yang tidak terdeteksi, atau praktik perjokian di mana individu lain mengerjakan ujian tanpa interaksi mencurigakan yang terekam antar akun dalam sistem) kemungkinan besar tidak akan terdeteksi oleh sistem ini.
4. **Implementasi dalam Modus *Offline* (Analisis Retrospektif):** Sistem deteksi yang dikembangkan dalam penelitian ini diimplementasikan dalam modus *offline*, yang berarti analisis dilakukan secara retrospektif terhadap data log yang telah terkumpul. Kemampuan untuk melakukan deteksi secara *real-time* dan memberikan peringatan langsung saat ujian berlangsung belum menjadi bagian dari lingkup penelitian ini.
5. **Konteks Institusional dan Generalisasi Model:** Data log yang digunakan berasal dari lingkungan Fasilkom UI. Karakteristik spesifik dari penggunaan Moodle, jenis ujian, kebijakan akademik, dan demografi mahasiswa di institusi lain mungkin berbeda. Oleh karena itu, kinerja model dapat bervariasi jika diterapkan secara langsung di institusi lain tanpa kalibrasi atau adaptasi ulang.
6. **Interpretabilitas Beberapa Komponen Model:** Meskipun analisis *feature importance* telah dilakukan untuk model seperti Random Forest, beberapa komponen dalam arsitektur *ensemble*, khususnya *Neural Network*, masih memiliki sifat inheren sebagai "kotak hitam" (*black box*), yang membuat interpretasi penuh terhadap logika pengambilan keputusannya menjadi lebih menantang.
7. **Cakupan Fitur yang Dihasilkan:** Meskipun 35 fitur awal telah diekstraksi dan direduksi menjadi 8 fitur stabil melalui VIF, ada kemungkinan fitur-fitur lain yang belum dieksplorasi dapat memberikan informasi tambahan yang berguna untuk deteksi kecurangan.

5.4 Saran

Berdasarkan temuan dan keterbatasan yang telah diidentifikasi dalam penelitian ini, serta mengacu pada kesenjangan penelitian yang telah diidentifikasi dalam tinjauan pustaka, berikut adalah beberapa saran untuk pengembangan dan penelitian selanjutnya:

- 1. Pengembangan Model dengan Data Riil Berlabel dan Pendekatan Semi-Supervised.**

Meskipun data artifisial terbukti berguna, melatih atau memvalidasi model dengan data riil yang memiliki label kecurangan terverifikasi akan meningkatkan kepercayaan dan generalisasi model. Mengingat kesulitan memperoleh data riil berlabel dalam skala besar, eksplorasi teknik *semi-supervised learning* atau *active learning* dapat dipertimbangkan untuk memanfaatkan data riil tak berlabel yang melimpah, sejalan dengan kerangka kerja yang diusulkan oleh Cen dkk. Cen et al. (2020) untuk deteksi anomali tanpa pengawasan dalam sistem *e-learning*.

- 2. Ekspansi Jenis Kecurangan yang Dapat Dideteksi.**

Penelitian mendatang dapat memperluas cakupan jenis kecurangan yang dideteksi dengan mengintegrasikan sumber data lain di luar log Moodle. Misalnya, analisis terhadap data dari *online proctoring tools*, analisis pola pengetikan (*keystroke dynamics*), atau bahkan analisis konten jawaban jika memungkinkan, dapat membantu mendeteksi bentuk kecurangan yang lebih beragam.

- 3. Implementasi Sistem Deteksi secara *Real-Time*.**

Mengembangkan sistem ini ke dalam modus operasional *real-time* akan memberikan manfaat yang lebih besar, karena memungkinkan intervensi atau peringatan dini dapat dilakukan saat ujian sedang berlangsung, bukan hanya analisis retrospektif. Implementasi ini dapat mengacu pada pendekatan yang telah dicontohkan oleh Moreno-Marcos dkk. Moreno-Marcos et al. (2023) dalam Statooodle yang menggunakan pendekatan *real-time monitoring*.

- 4. Validasi dan Adaptasi Model Lintas Institusi dan Konteks.**

Untuk meningkatkan generalisasi, model yang dikembangkan perlu diuji dan divalidasi pada dataset dari institusi pendidikan lain dengan karakteristik pengguna, mata kuliah, dan konfigurasi Moodle yang berbeda. Proses adaptasi atau *transfer learning* mungkin diperlukan.

- 5. Peningkatan Interpretabilitas Model Kompleks**

Meskipun analisis *feature importance* memberikan wawasan, penelitian lebih lanjut dapat mengeksplorasi teknik-teknik interpretabilitas *machine learning* yang lebih canggih (misalnya, SHAP, LIME) untuk memberikan penjelasan yang lebih detail mengenai bagaimana model, terutama *Neural Network*, membuat keputusan prediktif.

6. Integrasi dengan Sistem Peringatan Dini dan Intervensi Pedagogis.

Hasil deteksi sebaiknya tidak hanya digunakan untuk tujuan penindakan, tetapi juga diintegrasikan ke dalam sistem peringatan dini yang dapat memberikan umpan balik kepada mahasiswa atau dosen. Ini dapat menjadi dasar untuk intervensi pedagogis yang bertujuan meningkatkan kesadaran akan integritas akademik.

7. Kajian Aspek Etika, Privasi, dan Penerimaan Pengguna.

Implementasi sistem pemantauan otomatis seperti ini memerlukan kajian mendalam terkait aspek etika, perlindungan privasi data mahasiswa, dan persepsi serta penerimaan dari seluruh pemangku kepentingan (mahasiswa, dosen, dan administrator). Sebagaimana ditekankan oleh Gaevi dkk. Gaevi et al. (2015), sistem *learning analytics* yang efektif harus mempertimbangkan tidak hanya aspek teknis deteksi, tetapi juga implikasi pedagogis dan etis dari implementasi sistem tersebut.

8. Eksplorasi Teknik Deteksi Anomali yang Lebih Mendalam.

Selain model pembelajaran terawasi, penelitian selanjutnya dapat lebih fokus pada teknik deteksi anomali murni (*unsupervised anomaly detection*) sebagai komplementer, terutama untuk mengidentifikasi pola-pola kecurangan baru atau yang tidak terduga yang belum pernah ada dalam data pelatihan. Pendekatan ini dapat memperluas teknik yang telah didemonstrasikan oleh Alexandron dkk. Alexandron et al. (2019) dalam konteks MOOC menggunakan *Isolation Forest* dan *Local Outlier Factor*.

Dengan mempertimbangkan saran-saran tersebut, diharapkan penelitian di masa depan dapat menghasilkan sistem deteksi kecurangan akademik yang lebih canggih, adaptif, dan diterima secara luas, sehingga dapat berkontribusi lebih signifikan dalam menjaga integritas dan kualitas pendidikan tinggi di era digital. Hal ini sejalan dengan visi *learning analytics* sebagai bidang yang tidak hanya fokus pada deteksi masalah, tetapi juga pada pemahaman mendalam tentang pola perilaku belajar yang dapat membantu dalam pencegahan proaktif Ferguson (2012) dan mendukung proses pembelajaran yang adil serta mendorong integritas akademik melalui pendekatan yang konstruktif Gaevi et al. (2015).

BAB 6

PENUTUP

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, penyusunan laporan penelitian Skripsi yang berjudul "Pemantauan Kepatuhan secara Otomatis melalui Analisis Log pada Moodle Berbasis Kecerdasan Buatan" ini telah sampai pada bagian akhir. Seluruh rangkaian kegiatan penelitian, mulai dari identifikasi masalah, studi literatur, perancangan metodologi, implementasi sistem, hingga analisis hasil dan pembahasan, telah diuraikan secara komprehensif dalam bab-bab sebelumnya.

Pada Bab 5, telah dipaparkan secara rinci kesimpulan-kesimpulan utama yang ditarik dari keseluruhan hasil penelitian, keterkaitan temuan dengan tujuan dan pertanyaan penelitian yang telah dirumuskan, serta identifikasi keterbatasan-keterbatasan yang ada. Saran-saran untuk pengembangan dan penelitian selanjutnya juga telah disampaikan sebagai upaya untuk perbaikan dan eksplorasi lebih lanjut di masa mendatang.

Penulis berharap bahwa penelitian ini dapat memberikan kontribusi yang bermanfaat, baik secara teoretis bagi pengembangan ilmu pengetahuan di bidang kecerdasan buatan dan analisis data dalam konteks pendidikan, maupun secara praktis bagi institusi pendidikan dalam upaya menjaga dan meningkatkan integritas akademik di lingkungan pembelajaran daring. Semoga hasil penelitian ini dapat menjadi landasan bagi inovasi-inovasi selanjutnya dan memberikan inspirasi bagi peneliti lain yang memiliki minat serupa.

Akhir kata, penulis menyadari bahwa laporan penelitian ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun senantiasa diharapkan demi penyempurnaan di masa yang akan datang. Semoga laporan ini dapat memenuhi syarat sebagai karya ilmiah dan memberikan manfaat bagi semua pihak yang membacanya.

DAFTAR REFERENSI

- Aggarwal, C. C. (2017). *Outlier Analysis*. Springer, Cham, Switzerland, 2nd edition.
- Alexandron, G., Ruiprez-Valiente, J. A., and Pritchard, D. E. (2019). Towards a general purpose anomaly detection method to identify cheaters in massive open online courses. In *12th International Conference on Educational Data Mining (EDM 2019)*, pages 480–483, Montreal, Canada.
- Alsabhan, W. (2023). Student cheating detection in higher education by implementing machine learning and lstm techniques. *Sensors*, 23(8):4149.
- Balderas, A. and Caballero-Hernndez, J. A. (2020). Analysis of learning records to detect student cheating on online exams: Case study during covid-19 pandemic. In *8th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM)*, pages 752–757, Salamanca, Spain.
- Cen, H., Ruta, D., and Gabrys, B. (2020). A framework for unsupervised anomaly detection in e-learning systems. *Future Generation Computer Systems*, 102:837–850.
- Chang, S.-C. and Chang, K. L. (2023). Cheating detection of test collusion: A study on machine learning techniques and feature representation. *Educational Measurement: Issues and Practice*.
- Chirumamilla, A., Sindre, G., and Nguyen-Duc, A. (2020). Cheating in e-exams and paper exams: the perceptions of engineering students and teachers in norway. *Assessment & Evaluation in Higher Education*, 45(7):940–957.
- Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6):304–317.
- Gaevi, D., Dawson, S., and Siemens, G. (2015). Let's not forget: Learning analytics are about learning. *TechTrends*, 59(1):64–71.
- Huda, M., Jasmi, K. A., Zakaria, G. N., and et al. (2020). Challenges of rule-based academic dishonesty detection in online assessment. *International Journal of Emerging Technologies in Learning*, 15(4):95–106.
- Kamalov, F., Sulieman, H., and Calonge, D. S. (2021). Machine learning based approach to exam cheating detection. *PLoS ONE*, 16(7):e0254340.
- Lanier, M. M. (2006). Academic integrity and distance learning. *Journal of Criminal Justice Education*, 17(2):244–261.
- Mazza, R. and Dimitrova, V. (2004). Visualizing student tracking data to support instruc-

- tors in web-based distance education. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 154–161.
- Moreno-Marcos, P. M., Barredo, J., Muoz-Merino, P. J., and Delgado Kloos, C. (2023). Statoodle: A learning analytics tool to analyze moodle students' actions and prevent cheating. In *Lecture Notes in Computer Science*, volume 13884, pages 736–741. Springer.
- Murdoch, K. and House, D. (2019). Ghost in the shell: What happens when contract cheating meets online impersonation. In *ICAI Annual Conference*, New Orleans, LA.
- Nadeem, M., Kumar, V., Yunus, F., Jain, S., Aggarwal, D., and Syed, T. A. (2024). Revolutionizing financial fraud detection using advanced machine learning techniques. *Scientific Reports*, 14(1):28542.
- Niu, K., Zhang, J., Wang, Y., Wang, J., and Zhang, X. (2025). Effective machine learning methodology for medical prediction: a systematic review. *BMC Medical Informatics and Decision Making*, 25(1):15.
- Romero, C. and Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1355.
- Shatnawi, A., Al-Zoubi, A. Y., Faris, H., Eshtay, M., and Hassonah, M. A. (2024). E-exam cheating detection system for moodle lms. *Applied Sciences*, 14(1):397.
- Siemens, G. and Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5):30–32.
- Yulita, I. N., Hariz, F. A., Suryana, I., and Prabuwono, A. S. (2023). Educational innovation faced with covid-19: Deep learning for online exam cheating detection. *Education Sciences*, 13(2):194.
- Zhou, T. and Jiao, H. (2022). Data augmentation in machine learning for cheating detection in large-scale assessment. *Psychological Test and Assessment Modeling*, 64(4):425–444.
- Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. CRC Press, Boca Raton, FL.

LAMPIRAN

@todo

Silakan hapus lampiran ini ketika Anda mulai menggunakan *template*.

Template versi terbaru bisa didapatkan di <https://gitlab.com/ichlaffterlalu/latex-skripsi-ui-2017>. Daftar perubahan pada *template* hingga versi ini:

- versi 1.0.3 (3 Desember 2010):
 - *Template Skripsi/Tesis* sesuai ketentuan *formatting* tahun 2008.
 - Bisa diakses di <https://github.com/edom/uistyle>.
- versi 2.0.0 (29 Januari 2020):
 - *Template Skripsi/Tesis* sesuai ketentuan *formatting* tahun 2017.
 - Menggunakan BibTeX untuk sitasi, dengan format *default* sitasi IEEE.
 - *Template* kini bisa ditambahkan kode sumber dengan *code highlighting* untuk bahasa pemrograman populer seperti Java atau Python.
- versi 2.0.1 (8 Mei 2020):
 - Menambahkan dan menyesuaikan tutorial dari versi 1.0.3, beserta cara kontribusi ke template.
- versi 2.0.2 (14 September 2020):
 - Versi ini merupakan hasil *feedback* dari peserta skripsi di lab *Reliable Software Engineering* (RSE) Fasilkom UI, semester genap 2019/2020.
 - BibTeX kini menggunakan format sitasi APA secara *default*.
 - Penambahan tutorial untuk *longtable*, agar tabel bisa lebih dari 1 halaman dan header muncul di setiap halaman.
 - Menambahkan tutorial terkait penggunaan BibTeX dan konfigurasi *header/footer* untuk pencetakan bolak-balik.
 - Label "Universitas Indonesia" kini berhasil muncul di halaman pertama tiap bab dan di bagian abstrak - daftar kode program.
 - *Hyphenation* kini menggunakan *babel* Bahasa Indonesia. Aktivasi dilakukan di *hyphen-indonesia.tex*.
 - Minor adjustment untuk konsistensi *license* dari template.
- versi 2.0.3 (15 September 2020):

- Menambahkan kemampuan orientasi *landscape* beserta tutorialnya.
 - \captionsource telah diperbaiki agar bisa dipakai untuk longtable.
 - Daftar lampiran kini telah tersedia, lampiran sudah tidak masuk daftar isi lagi.
 - Nomor halaman pada lampiran dilanjutkan dari halaman terakhir konten (daftar referensi).
 - Kini sudah bisa menambahkan daftar isi baru untuk jenis objek tertentu (custom), seperti: "Daftar Aturan Transformasi". Sudah termasuk mekanisme *captioning* dan tutorialnya.
 - Perbaikan minor pada tutorial.
- versi 2.1.0 (8 September 2021):
 - Versi ini merupakan hasil *feedback* dari peserta skripsi dan tesis di lab *Reliable Software Engineering* (RSE) Fasilkom UI, semester genap 2020/2021.
 - Minor edit: "Lembar Pengesahan", dsb. di daftar isi menjadi all caps.
 - Experimental multi-language support (Chinese, Japanese, Korean).
 - *Support* untuk justifikasi dan word-wrapping pada tabel.
 - Penggunaan suffix "(sambungan)" untuk tabel lintas halaman. Tambahan support suffix untuk \captionsource.
 - versi 2.1.1 (7 Februari 2022):
 - Update struktur mengikuti fork template versi 1.0.3 di <https://github.com/rkkautsar/edom/ui-thesis-template>.
 - *Support* untuk simbol matematis amsfonts.
 - Kontribusi komunitas terkait improvement GitLab CI, atribusi, dan format sitasi APA bahasa Indonesia.
 - Perbaikan tutorial berdasarkan perubahan terbaru pada versi 2.1.0 dan 2.1.1.
 - versi 2.1.2 (13 Agustus 2022):
 - Modifikasi penamaan beberapa berkas.
 - Perbaikan beberapa halaman depan (halaman persetujuan, halaman orisinalitas, dsb.).
 - *Support* untuk lembar pengesahan yang berbeda dengan format standar, seperti Laporan Kerja Praktik dan Disertasi.
 - Kontribusi komunitas terkait kesesuaian dengan format Tugas Akhir UI, kelengkapan dokumen, perbaikan format sitasi, dan *quality-of-life*.
 - Perbaikan tutorial.
 - versi 2.1.3 (22 Februari 2023):

- Dukungan untuk format Tugas Akhir Kelompok di Fasilkom UI.
- Dukungan untuk format laporan Kampus Merdeka Mandiri di Fasilkom UI.
- Minor *bugfix*: Perbaikan kapitalisasi variabel.
- Quality-of-Life: Pengaturan kembali config/settings.tex.
- Tutorial untuk beberapa *use case*.
- versi 2.2.0 (28 Agustus 2024):
 - Perbaikan format agar sesuai dengan format Tugas Akhir terbaru. Hal ini mencakup halaman judul, halaman pernyataan orisinalitas, header/footer, dan lampiran.
- versi 2.2.1 (16 Desember 2024):
 - *Bugfix*: isu *header* dan *footer* untuk halaman bolak-balik.
 - *Bugfix*: isu *auto-wrapping* pada kode yang tidak bisa berjalan sejak v2.2.0.
 - *Bugfix*: isu penomoran objek kustom yang tidak sesuai konvensi [bab].[objek].
 - *Bugfix*: penomoran bab di Daftar Isi yang belum sesuai konvensi Tugas Akhir UI.
 - *Bugfix*: hal-hal lain pada *formatting* sesuai dengan permintaan dari Perpustakaan Fasilkom UI.
 - Perbaikan *formatting* untuk *landscape* dengan *library pdflscape*.
 - Perbaikan cara memasukkan sebuah persamaan ke dalam daftar persamaan.
 - Perbaikan penggunaan "saya" menjadi "kami" untuk dokumen-dokumen awal pada Tugas Akhir Kelompok.
 - Fitur baru: *Support* untuk *code highlighting* pada berbagai bahasa pemrograman yang tidak di-*support* secara *default* oleh *library listings*.
 - Fitur baru: *Support* untuk *glossary* (daftar istilah).
 - Perbaikan *major* pada tutorial, termasuk menampilkan contoh kode ke dalam PDF tutorial, dan pengaturan ulang subbab.

Lampiran hadir untuk menampung hal-hal yang dapat menunjang pemahaman terkait tugas akhir, namun akan mengganggu *flow* bacaan sekiranya dimasukkan ke dalam bacaan. Lampiran bisa saja berisi data-data tambahan, analisis tambahan, penjelasan istilah, tahapan-tahapan antara yang bukan menjadi fokus utama, atau pranala menuju halaman luar yang penting.

Subbab dari Lampiran 2

@todo

Isi subbab ini sesuai keperluan Anda. Anda bisa membuat lebih dari satu judul lampiran, dan tentunya lebih dari satu subbab.